THE MOST COMPLETE STARTER KIT

TUTORIAL FOR UNO

- EN FRANCAIS -

V1.0.17.7.10

Préface

Notre companie

Etablie en 2001, Elegoo inc. est une entreprise de technologie qui s'efforce de fournir des contenus open source pour la recherche, l'industrie, mais aussi l'éducation.

Nos locaux sont à Shenzhen, qui est connue comme la "Silicon Valley" chinoise. Nous sommes plus de 150 employés.

Notre production s'étend des câbles de connexion, cartes UNO jusqu'aux starters kits, conçus pour tous les niveaux de savoir. Nous produisons aussi des accessoires pour les cartes Raspberry comme les écrans TFT 2.8'' et les STM32.

Dans le futur, nous envisageons de nous investir dans le domaine des imprimantes 3D.

Tous nos produits sont conformes aux normes de qualité internationales et sont reconnus sur les marchés où ils sont commercialisés.

Notre site web: http://www.elegoo.com

Nos boutiques Amazon :

US Amazon storefront: http://www.amazon.com/shops/A2WWHQ25ENKVJ1

CA Amazon storefront: http://www.amazon.ca/shops/A2WWHQ25ENKVJ1

UK Amazon storefront: http://www.amazon.co.uk/shops/AZF7WYXU5ZANW

DE Amazon storefront: http://www.amazon.de/shops/AZF7WYXU5ZANW

FR Amazon storefront: http://www.amazon.fr/shops/AZF7WYXU5ZANW

ES Amazon storefront: http://www.amazon.es/shops/AZF7WYXU5ZANW

IT Amazon storefront: http://www.amazon.it/shops/AZF7WYXU5ZANW

Ce tutoriel

Ce tutoriel est conçu pour les débutants. Vous y apprendrez les informations essentielles sur la manière d'utiliser une carte UNO. Mais si vous souhaitez allez plus loin, nous vous conseillons de faire acquisition de livres comme il y en a plusieurs références en vente sur des sites comme Amazon.

Beaucoup de codes contenus dans ce tutoriel sont de Simon Monk qui est un auteur reconnu de livre concernant l'Arduino.

Service consommateur

Nous nous efforçons de produire nos kits avec le plus grand soin et un souci permanent de qualité.

Nous sommes à votre écoute. N'hésitez pas à prendre contact avec nous à l'adresse suivante : service@elegoo.com ou EUservice@elegoo.com.

Nous sommes impatients de lire vos remarques et/ou suggestions.

Nous mettrons tout en œuvre pour qu'un ingénieur expérimenté vous réponde dans les 12h ou 24h pendant les périodes de congés.

Packing list

www.elegoo.com

Servo Motor (SG90) 1PC

ULN2003 Stepper Motor Driver Module 1PC

Power Supply Module 1PC

GY-521 Module 1PC

Elegoo UNO R3 Controller Board 1PC

LCD 1602 Module (with pin header) 1PC

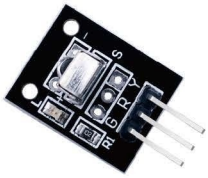
Rc522 RFID Module 1PC

Stepper Motor 1PC

Prototype Expansion Module 1PC

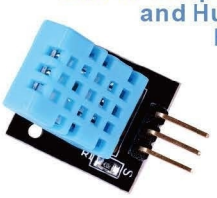
Contact us : service@elegoo.com

IR Receiver Module 1PC

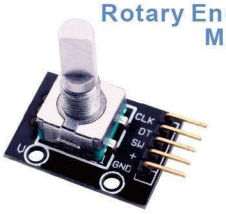


Joystick Module 1PC

DHT11 Temperature and Humidity Module 1PC



Rotary Encoder Module 1PC



Ultrasonic Sensor 1PC

DS1307 RTC Module 1PC

HC-SR501 PIR Motion Sensor Module 1PC

Sound Sensor Module 1PC

Water Level Detection Sensor Module 1PC

www.elegoo.com

1 Digit 7-Segment Display 1PC

4 Digit 7-Segment Display 1PC

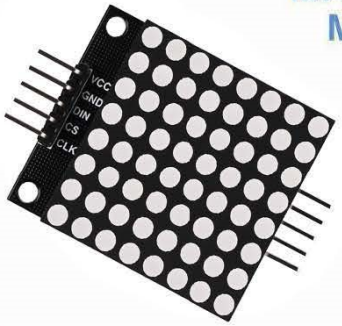
74HC595 IC 1PC

L293D 1PC

Potentiometer (10K) 2PCS

Remote 1PC

MAX7219 Module 1PC

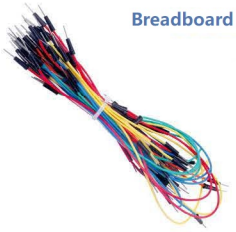


Active Buzzer 1PC

Passive Buzzer 1PC

Contact us : service@elegoo.com

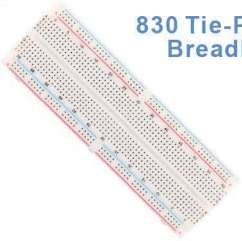
Breadboard Jumper Wire
65PCS



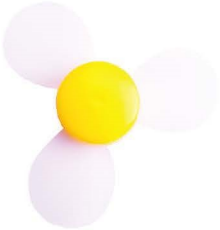
Female-to-Male Dupont Wire
20PCS

USB Cable
1PC

830 Tie-Points Breadboard
1PC



Fan Blade and 3-6VDC Motor (with wire) each
1PC



Membrane Switch Module
1PC

9V Battery With DC
1PC

5V Relay
1PC

9V 1A Adapter
1PC

Contact us : service@elegoo.com

www.elegoo.com

Electrolytic Capacitor
(100uF 50V)
2PCS

Electrolytic Capacitor
(10uF 50V)
2PCS

NPN Transistor
(S8050)
5PCS

NPN Transistor
(PN2222)
5PCS

Resistor
120PCS

Button (Small)
5PCS

Thermistor
1PC

Diode Rectifier
5PCS

Tilt Ball Switch
1PC

Contact us : service@elegoo.com

www.elegoo.com

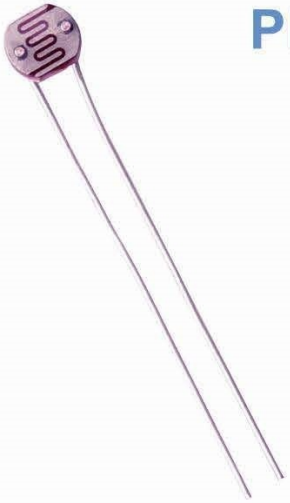
Red LED
5PCS

Yellow LED
5PCS

Blue LED
5PCS

Green LED
5PCS

Photoresistor
(Photocell)
2PCS



RGB LED
2PCS

White LED
5PCS

22pF Ceramic
Capacitor
5PCS

104pF Ceramic
Capacitor
5PCS

Contact us : service@elegoo.com

Content

Leçon 0 Installation de l'environnement de programmation 12

Leçon 1 Ajouter une bibliothèque / utiliser le moniteur série....23

Leçon 2 Blink 32

Leçon 3 LED 42

Leçon 4 RGB LED 49

Leçon 5 Digital Inputs 58

Leçon 6 Active buzzer 63

Leçon 7 Passive Buzzer 67

Leçon 8 Tilt Ball Switch 71

Leçon 9 Servomoteur....75

Leçon 10 Ultrasonic Sensor Module....79

Leçon 11 Membrane Switch Module 84

Leçon 12 DHT11 Temperature and Humidity Sensor 90

Leçon 13 Analog Joystick Module 96

Leçon 14 IR Receiver Module 101

Leçon 15 MAX7219 LED Dot Matrix Module....107

Leçon 16 GY-521 Module....111

Leçon 17 HC-SR501 PIR Sensor....119

Exemple de la piste de dance....124

Leçon 18 Water Level Detection Sensor Module....128

Leçon 19 Real Time Clock Module....133

Leçon 20 Sound Sensor Module 138

Leçon 21 RC522 RFID Module....144

Leçon 22 LCD Display 149

Leçon 23 Thermometer 154

Leçon 24 Eight LED with 74HC595 159

Leçon 25 The Serial Monitor....166

Leçon 26 Photocell....171

Leçon 27 74HC595 And Segment Display.....176

Leçon 28 Four Digital Seven Segment Display.....182

Leçon 29 DC Motors.....187

Leçon 30 Relay.....196

Leçon 31 Stepper Motor.....201

Leçon 32 Controlling Stepper Motor With Remote.....209

Leçon 33 Controlling Stepper Motor With Rotary Encoder.....213

Leçon 0 Installation de l'environnement de programmation

Introduction

L'IDE Arduino (Arduino Integrated Development Environment (ou IDE)) est un logiciel dédié de la plateforme Arduino.

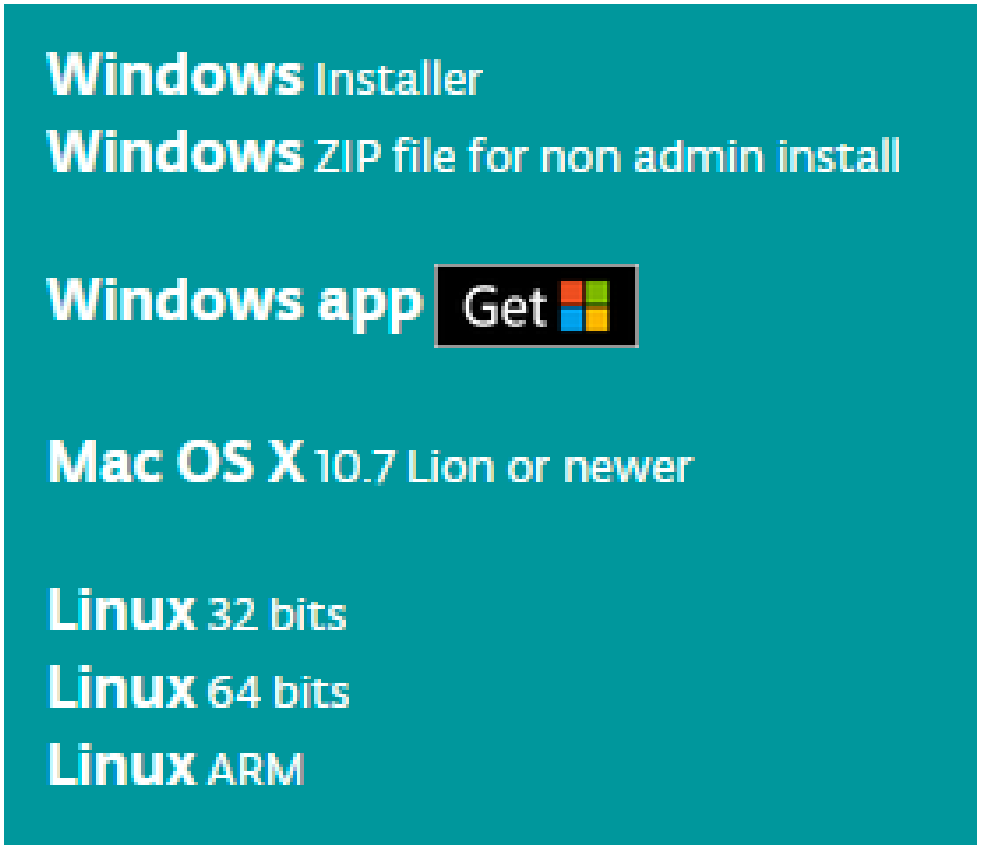
Dans cette leçon, vous allez procéder à l'installation de celui-ci.

Le logiciel est disponible pour Windows, MAC, LINUX.

STEP 1: Rendez-vous sur https://www.arduino.cc/en/Main/Software

Téléchargez la dernière version disponible (qui n'est plus forcément celle présente sur la capture d'écran).

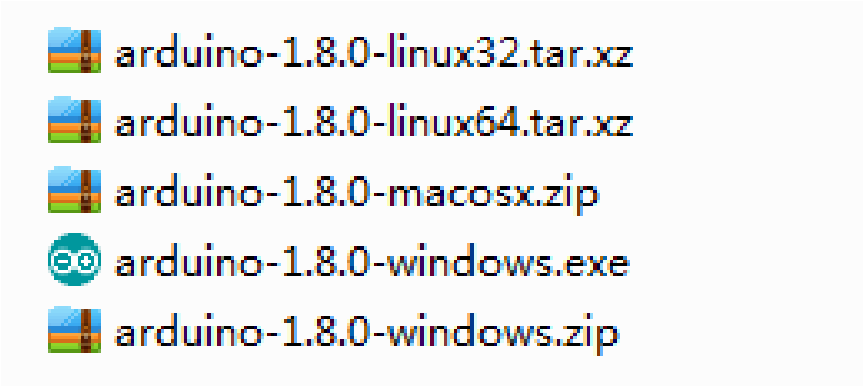
STEP2：Téléchargez la version correspondant à votre ordinateur



Si vous souhaitez faire un don pour soutenir la communauté, cliquez sur “contribute…”

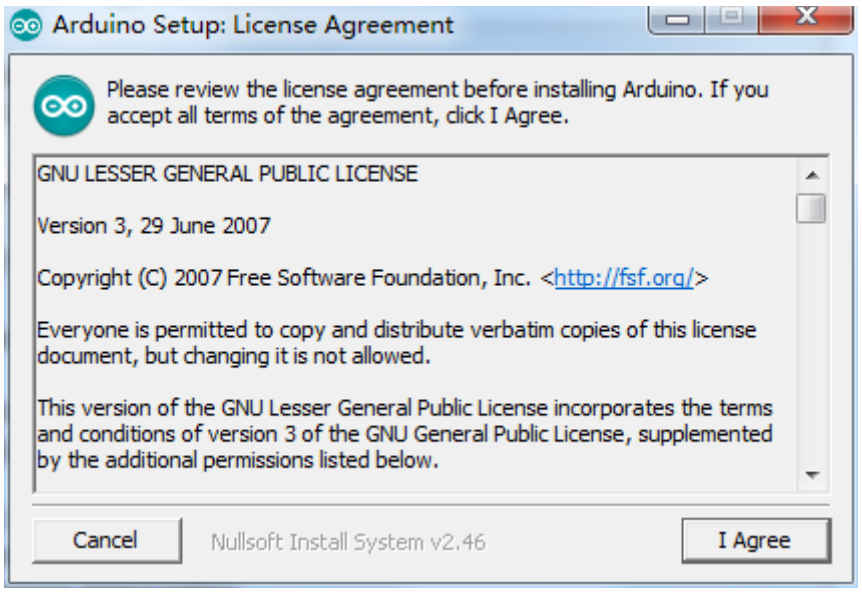
Si vous souhaitez téléchargez simplement, cliquez sur “just download”.

Sélectionnez le fichier correspondant à votre environnement de travail.

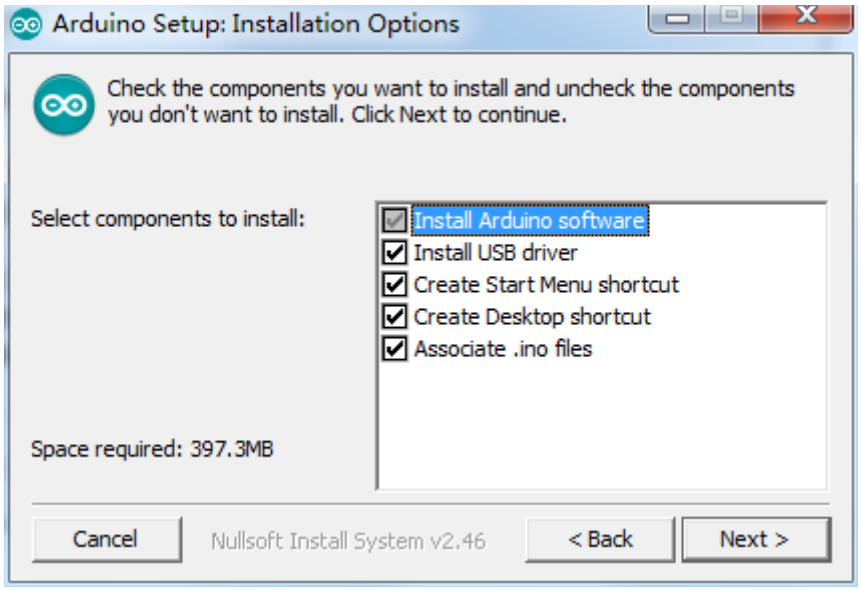


Installation sous Windows

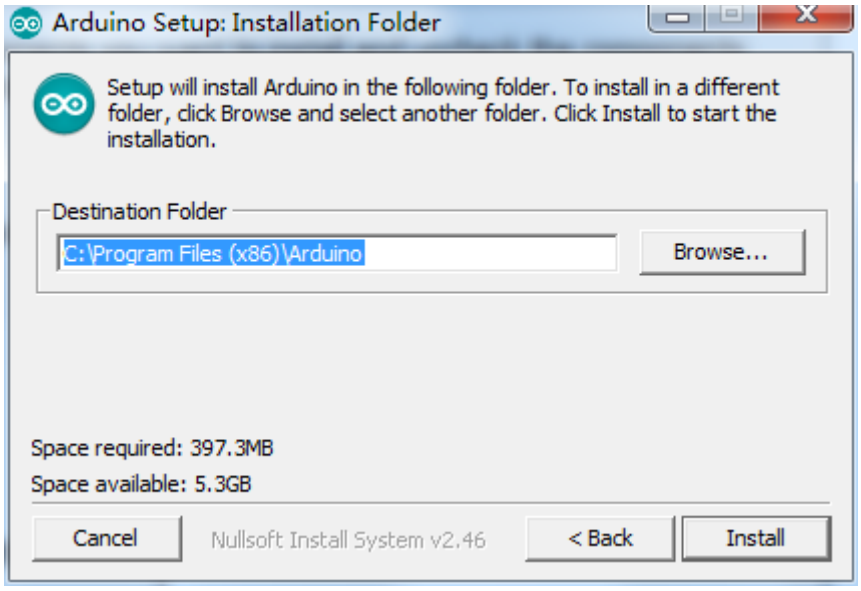
Exécutez le fichier.



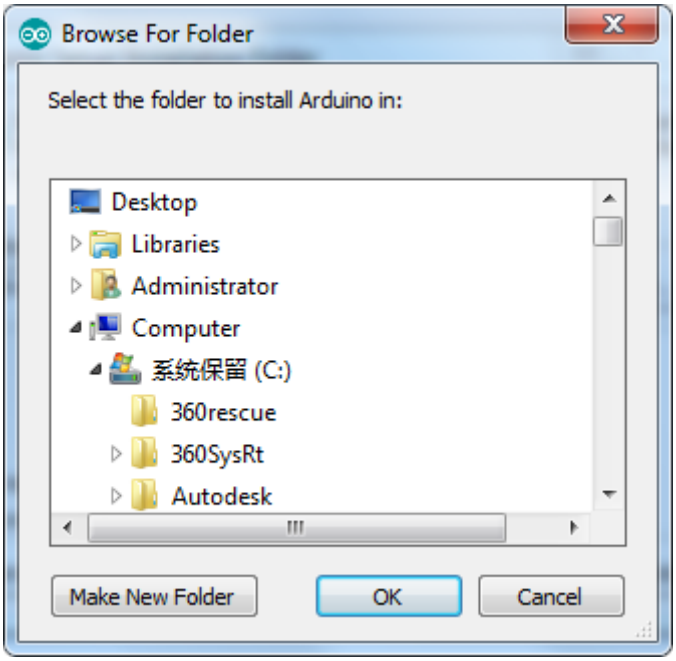
Cliquez sur *I Agree*



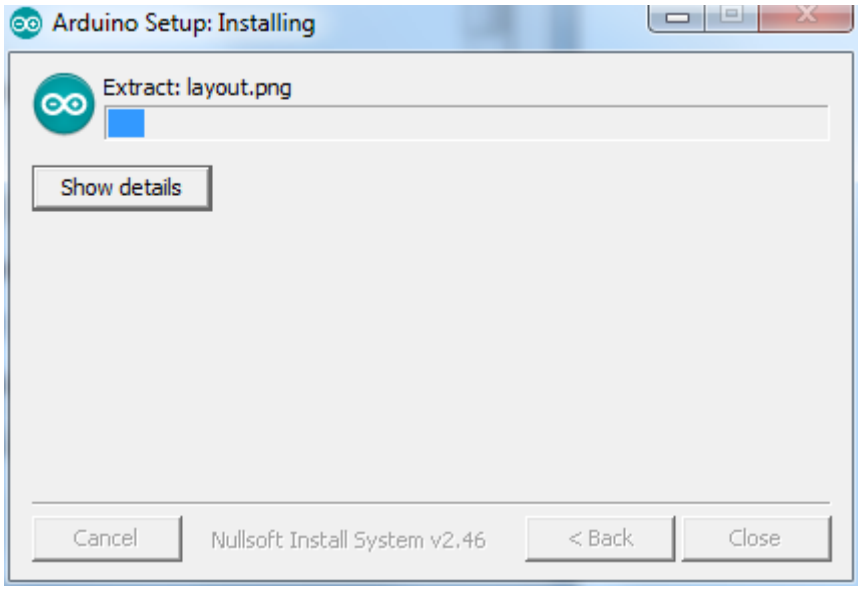
Cliquez *Next*



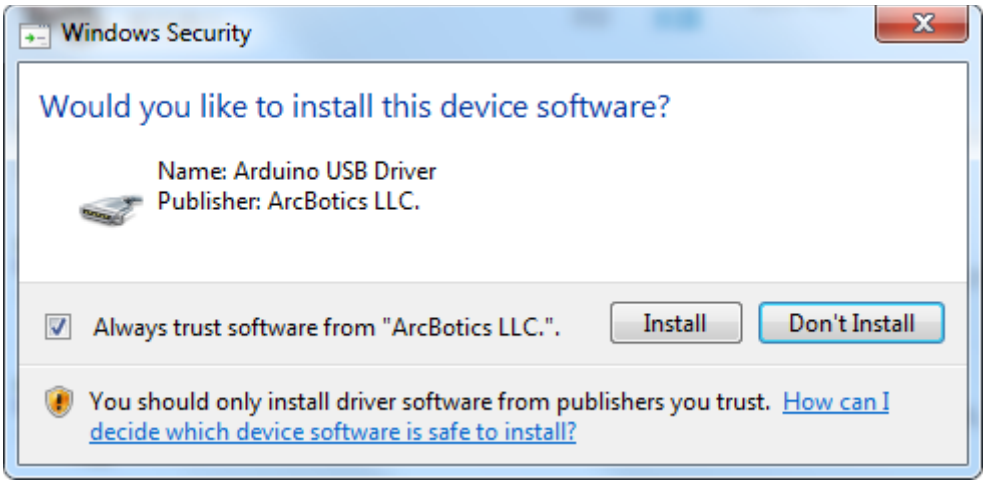
Cliquez sur *Browse* si vous souhaitez définir un autre répertoire.



Cliquez sur *Install*

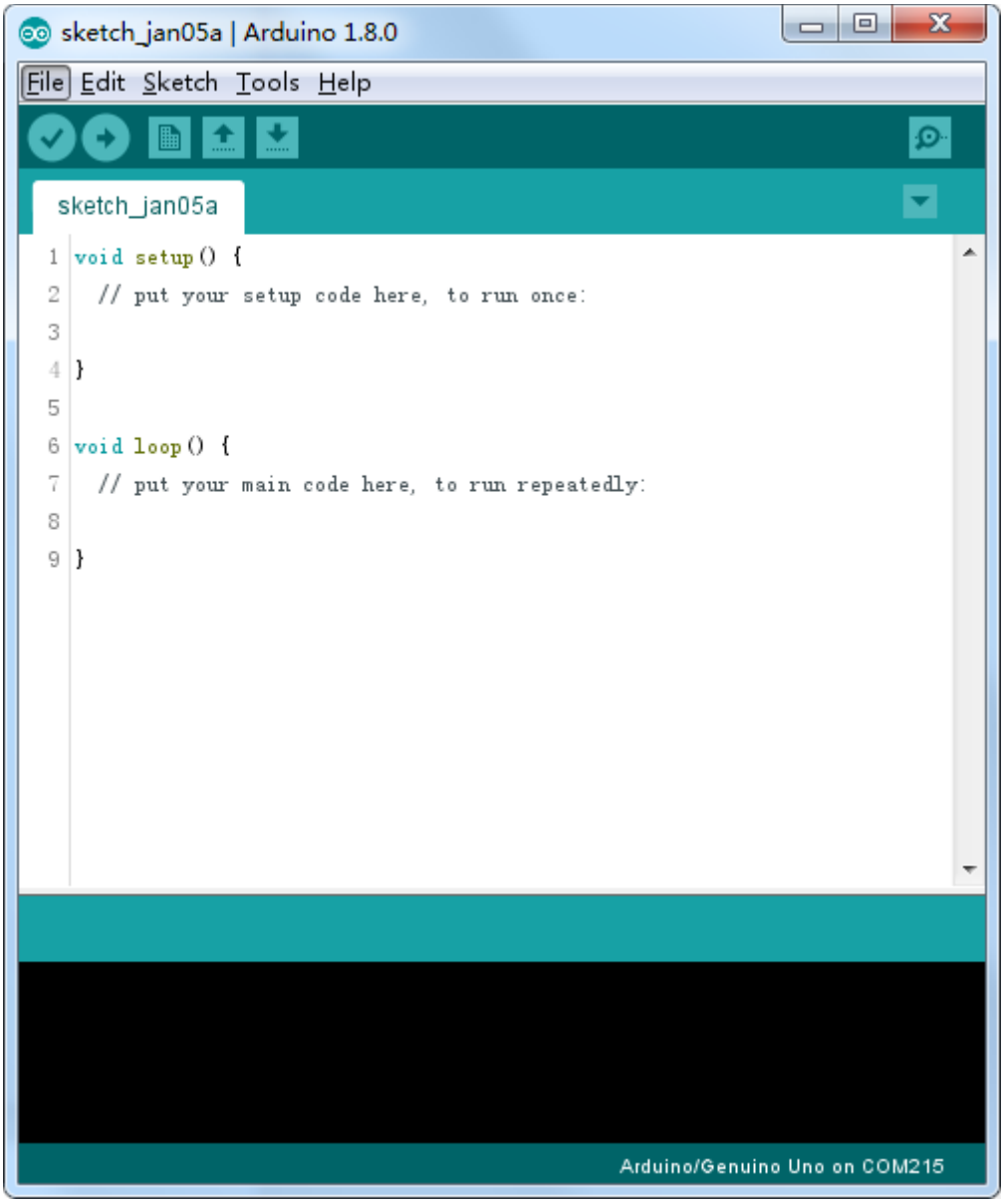


Enfin, cliquez de nouveau sur *Install*.



L'icône suivant apparait sur votre bureau

Double cliquez pour lancer l'IDE

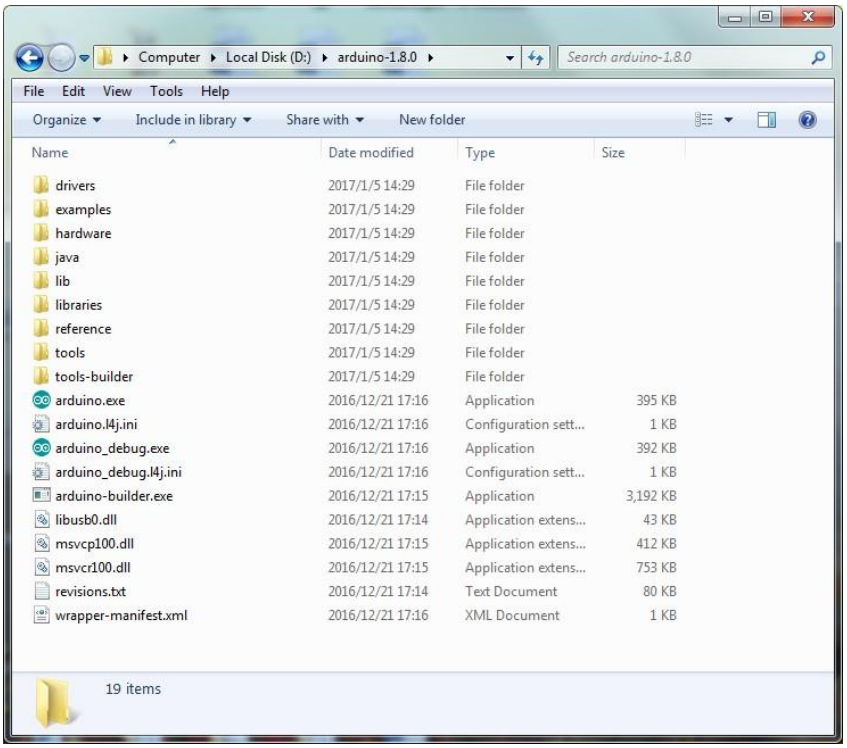


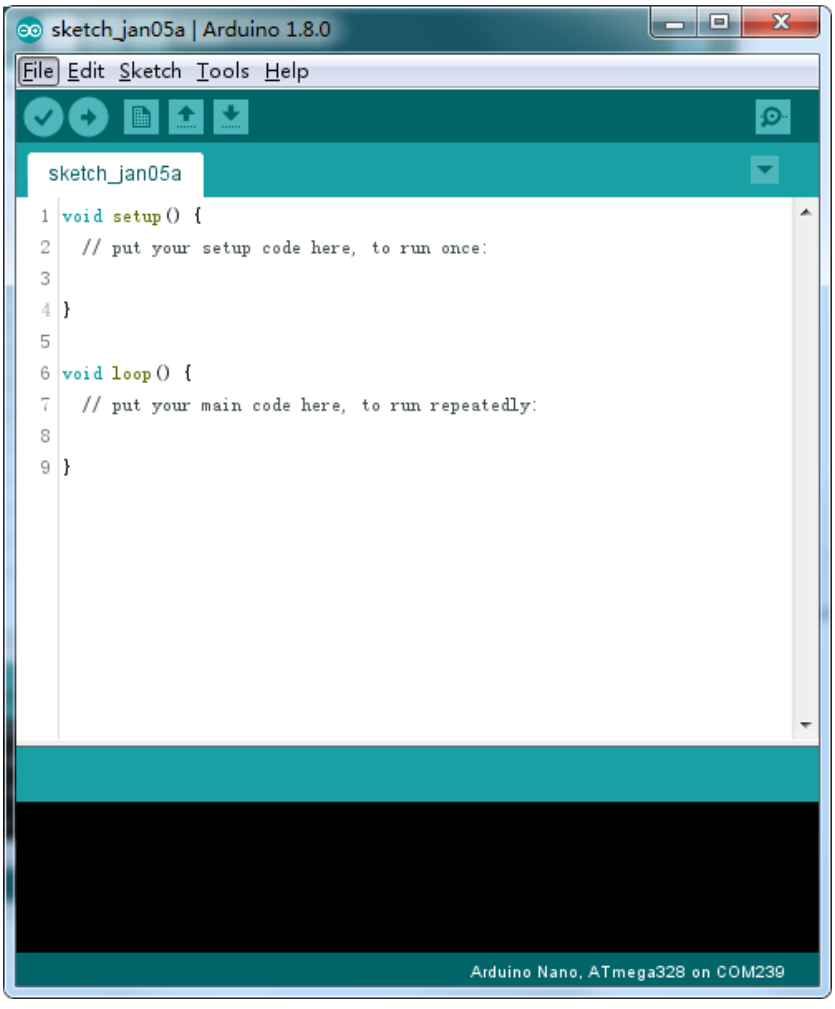
Ceci est la fenêtre de départ. Elle contient l'ossature de code commune et obligatoire.

Pour installer le logiciel depuis un fichier zip, veuillez suivre les instructions suivantes.

Téléchargez le fichier zip. Faites l'extraction des fichiers dans un dossier.

Lancer le fichier *arduino.exe*



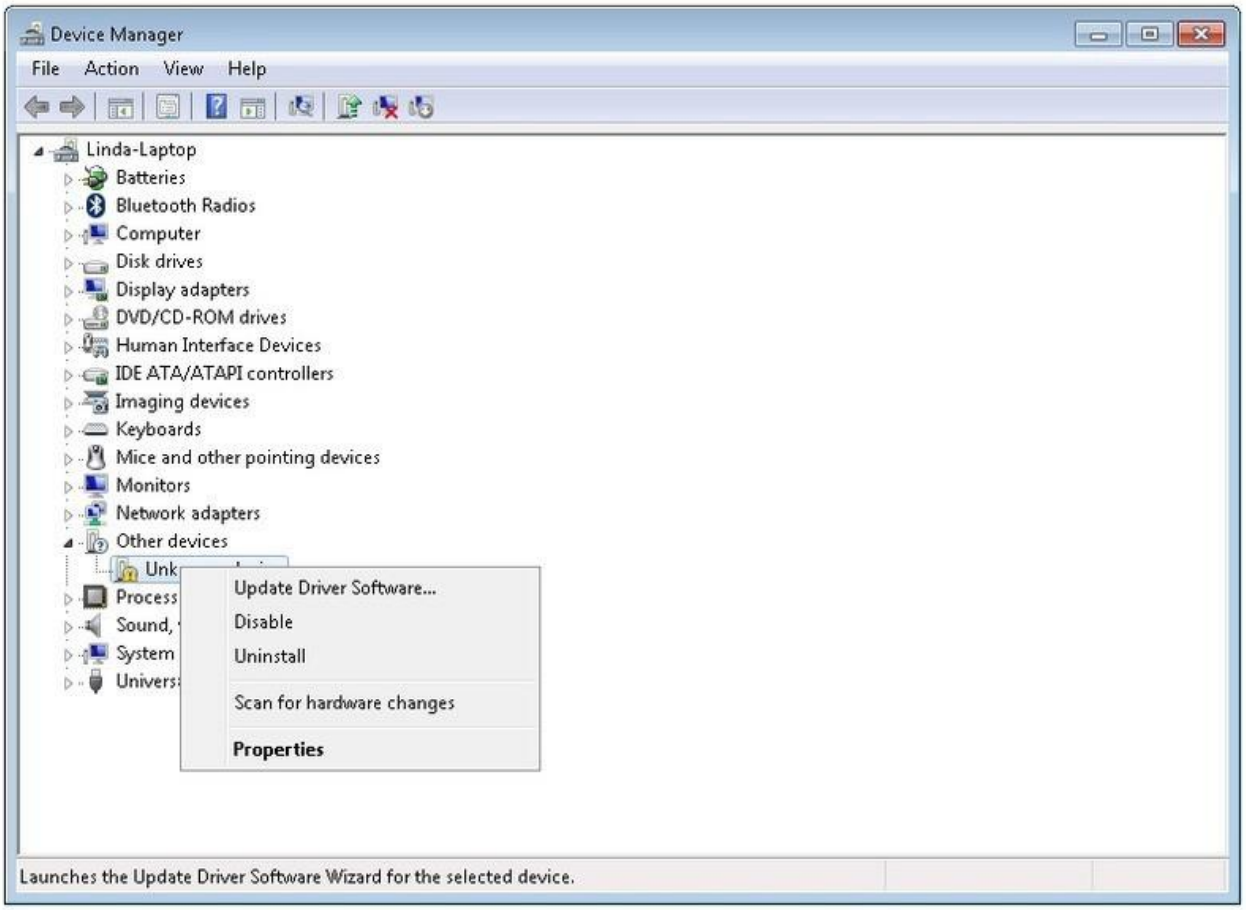


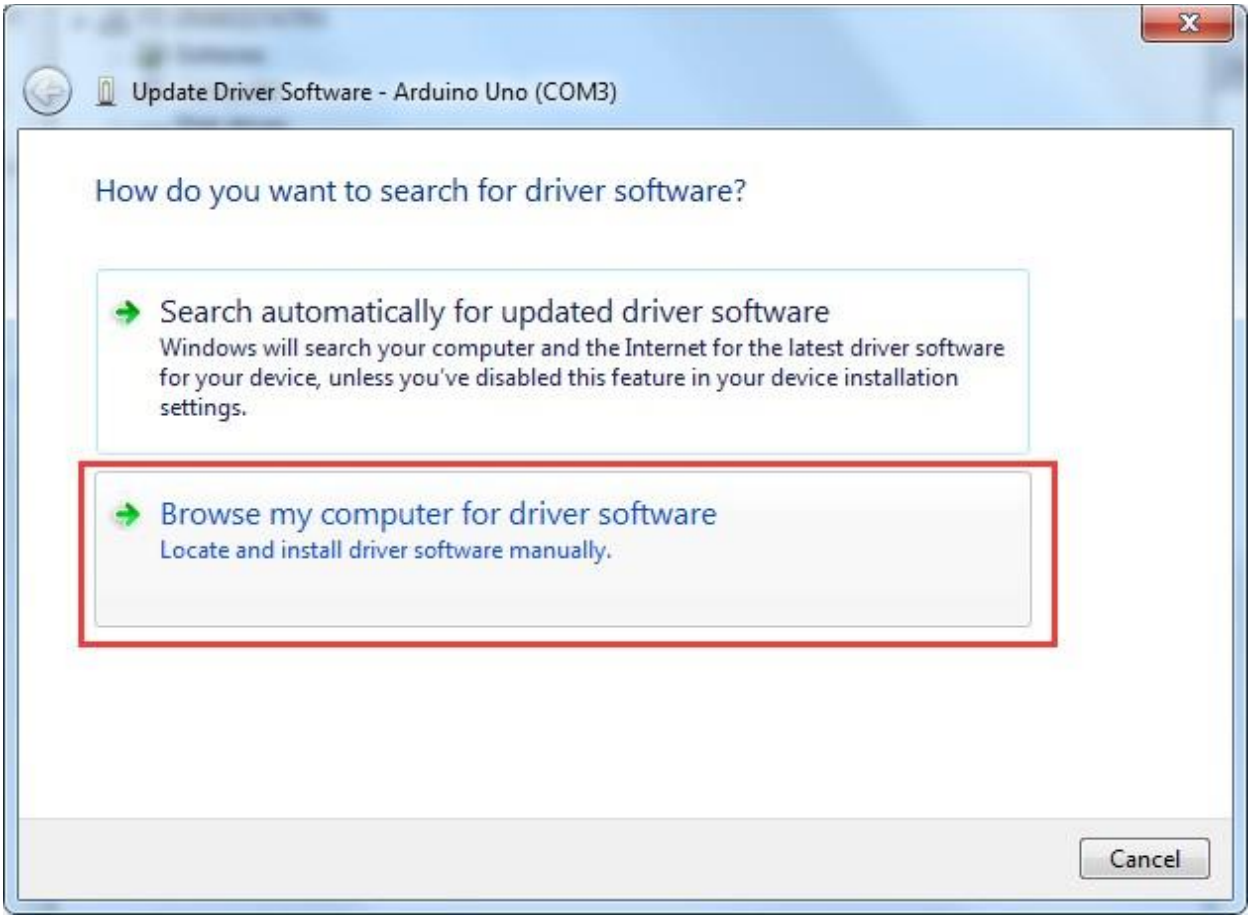
Si vous procédez de cette dernière manière, il faut installer les drivers de la carte.

Le répertoire dézippé contient à la fois les fichiers nécessaires au bon fonctionnement de l'IDE, mais aussi ceux nécessaires à l'installation des drivers USB de la carte UNO.

Branchez une carte UNO à un port USB de votre ordinateur. Vous allez certainement voir un message apparaitre mentionnant que Windows a découvert un nouveau matériel. **Ignorez ce message et fermez les tentatives de Windows de faire l'installation.**

La méthode la plus fiable est d'aller dans le gestionnaire de périphérique et de faire l'installation manuellement.





Faites un clic droit et sélectionnez le menu "*Mettre à jour le driver*". Sélectionnez ensuite *parcourir* et allez chercher le répertoire dans lequel vous avez dézippé le fichier Arduino.

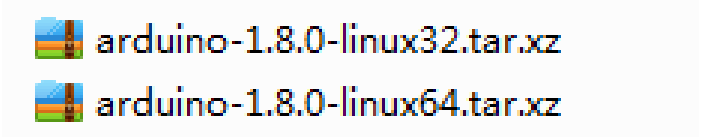
Cliquez '*Suivant*'. Après un message de sécurité, vous obtenez l'écran de confirmation de la bonne installation.

Installation sous Mac OS X

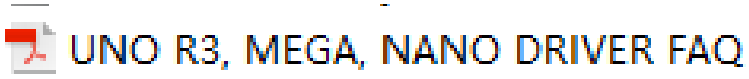
Téléchargez et dézippez le fichier suivant:

Installation Linux

Sélectionnez le fichier correspondant à votre environnement de travail



Conseil : vous trouverez des conseils et astuces, des réponses aux questions fréquentes :



Leçon 1 Ajouter une bibliothèque / utiliser le moniteur série

Installer des bibliothèques complémentaires

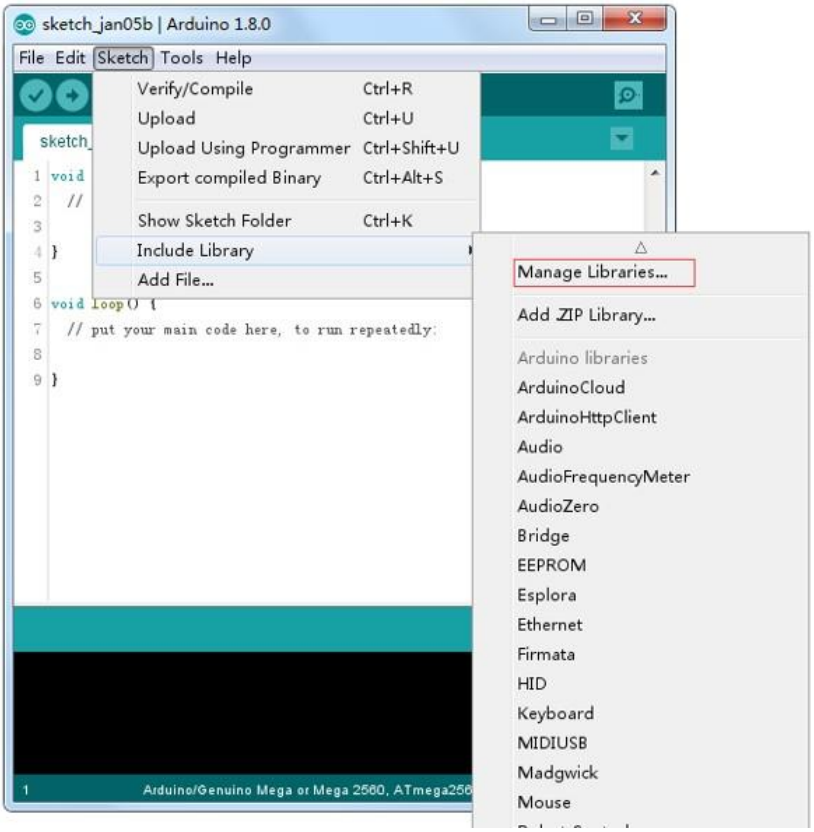
Lorsque vous aurez bien saisi les fonctions intégrées de l'environnement Arduino, vous aurez certainement le besoin ou l'envie d'aller encore plus loin, avec pourquoi pas des bibliothèques complémentaires.

Qu'est-ce qu'une bibliothèque?

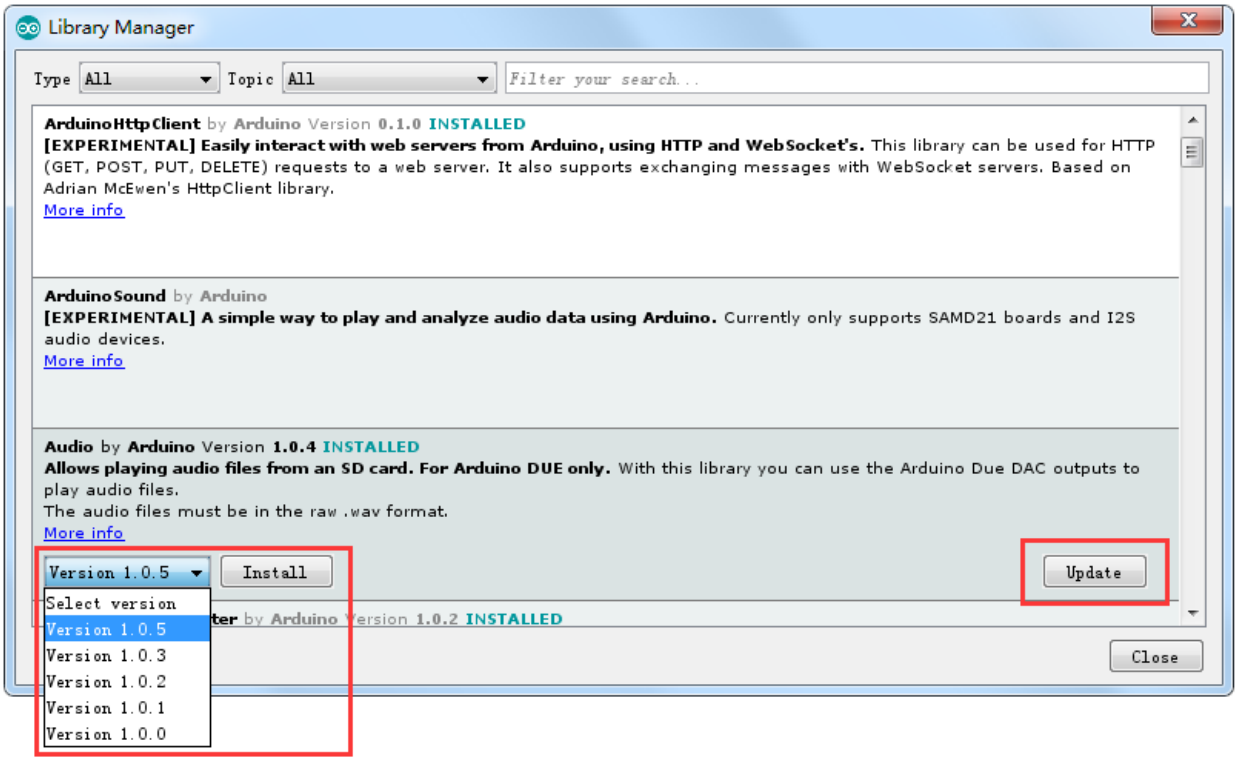
Une bibliothèque est une suite d'instructions qui rendent beaucoup plus facile l'utilisation de composants complexes. Cela peut être pour l'utilisation d'un écran à cristaux liquide, pour l'utilisation d'un servomoteur etc…

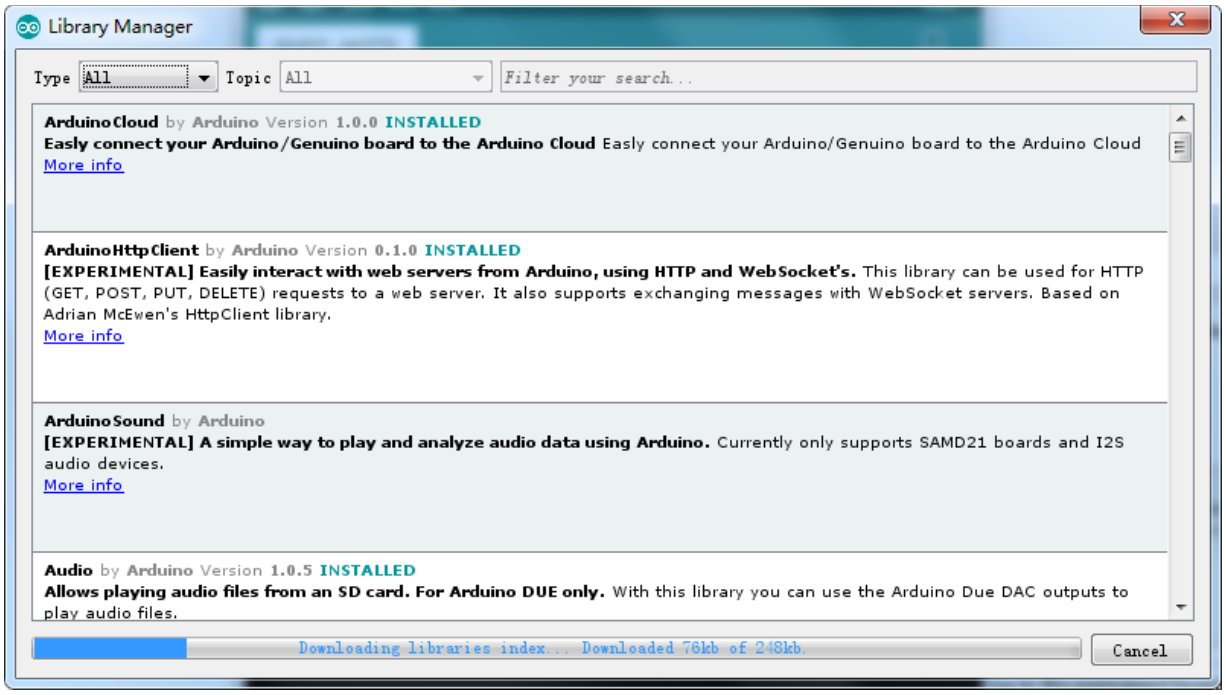
Comment installer

L'IDE contient un centre de management des bibliothèques qui permet de vérifier la bonne installation de telle ou telle bibliothèque ou d'en ajouter de nouvelles
Pour installer une nouvelle bibliothèque, cliquez sur le menu suivant :

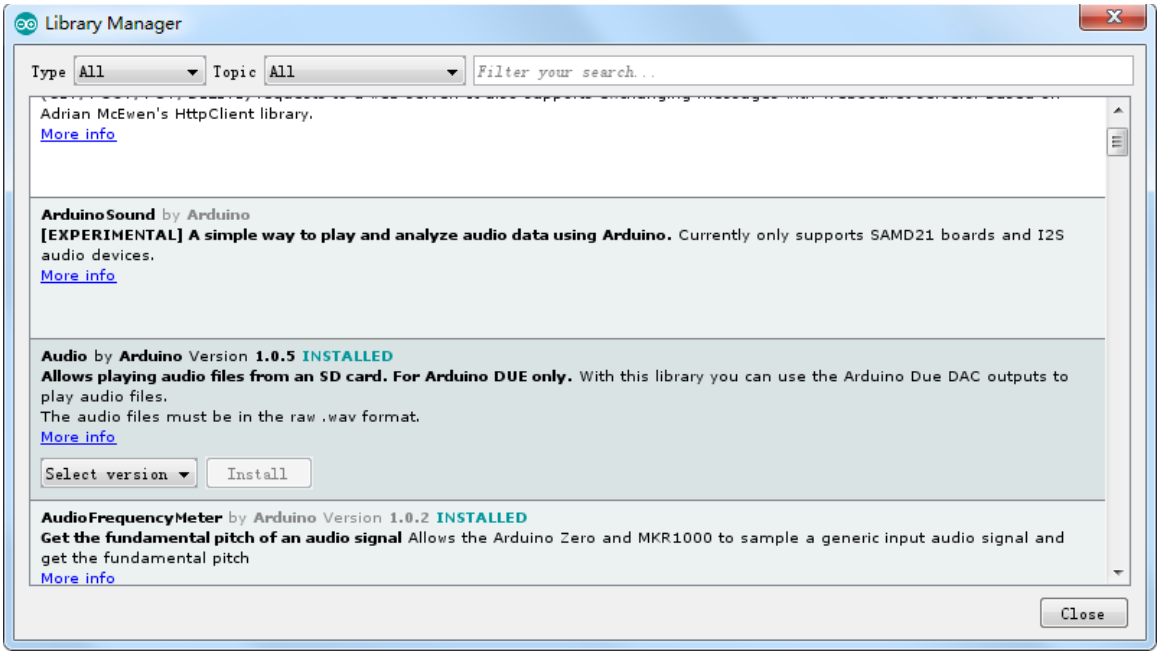


Vous allez pouvoir voir l'ensemble des bibliothèques déjà présentes ainsi que la version utilisée :





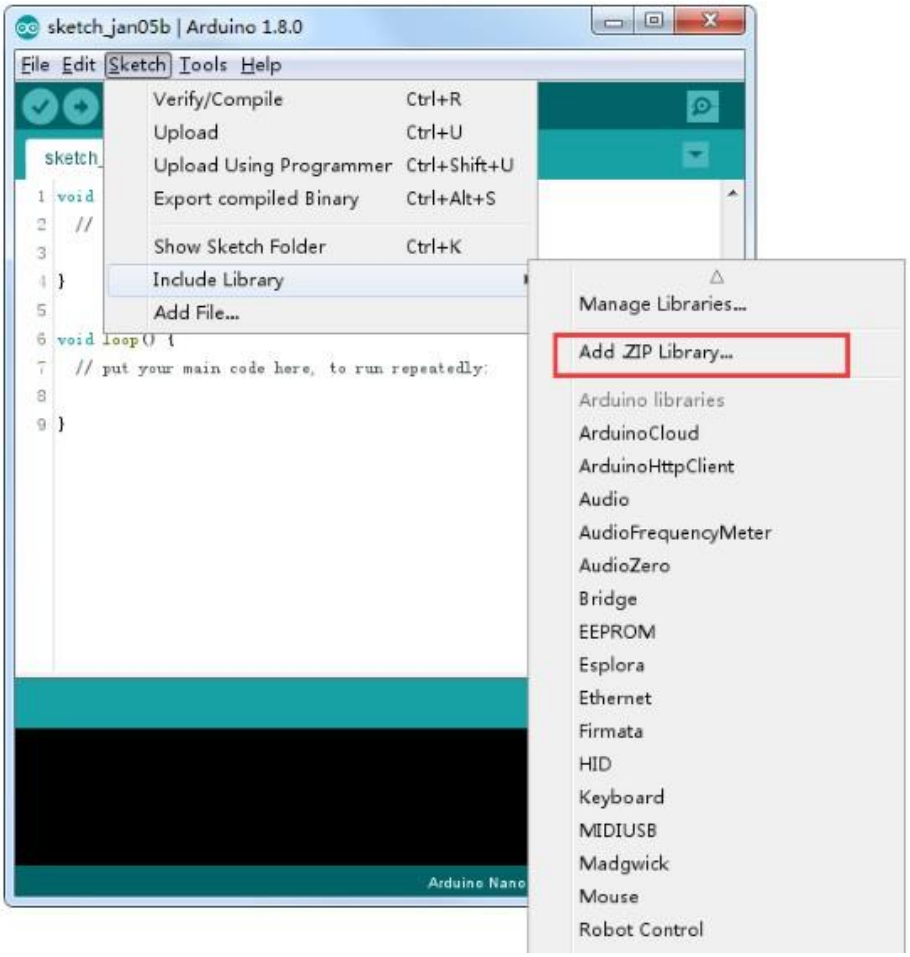
Si la dernière version d'une bibliothèques n'est pas installée, le bouton "Installer" se dégrise, vous pouvez procéder à son installation.



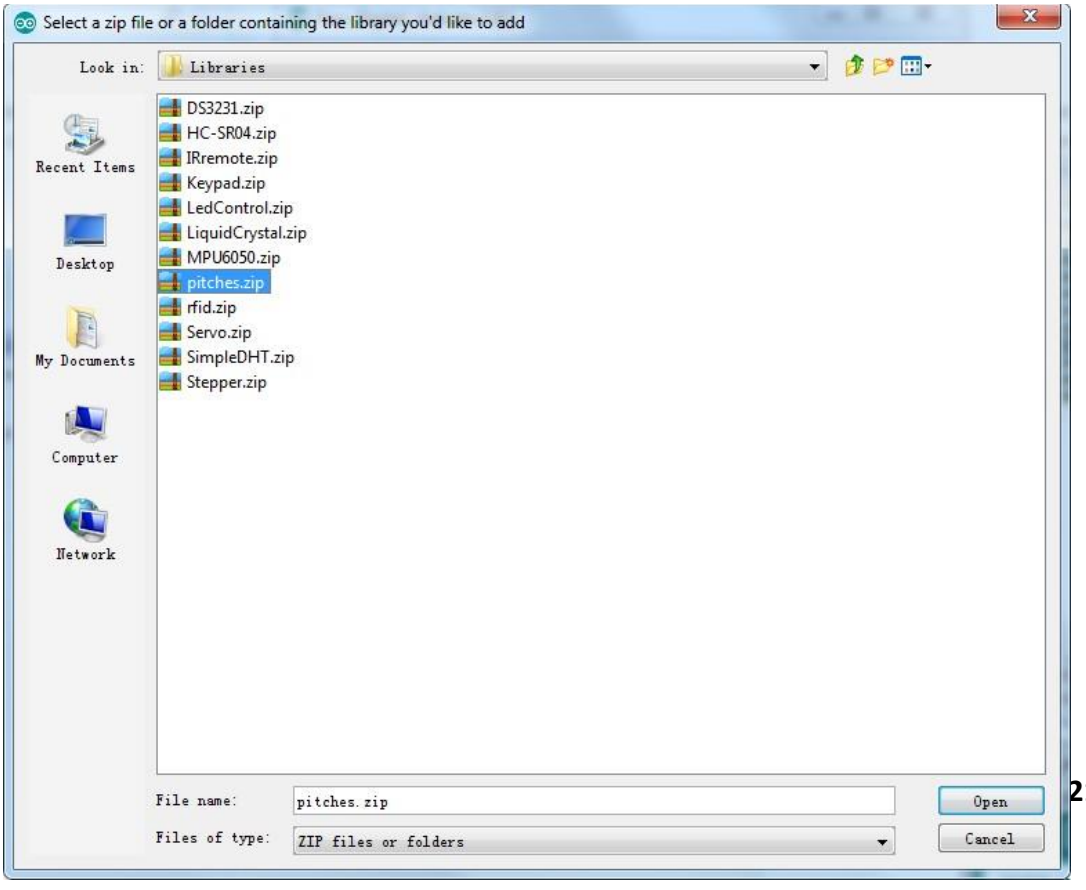
Si la bibliothèque recherchée n'apparait pas

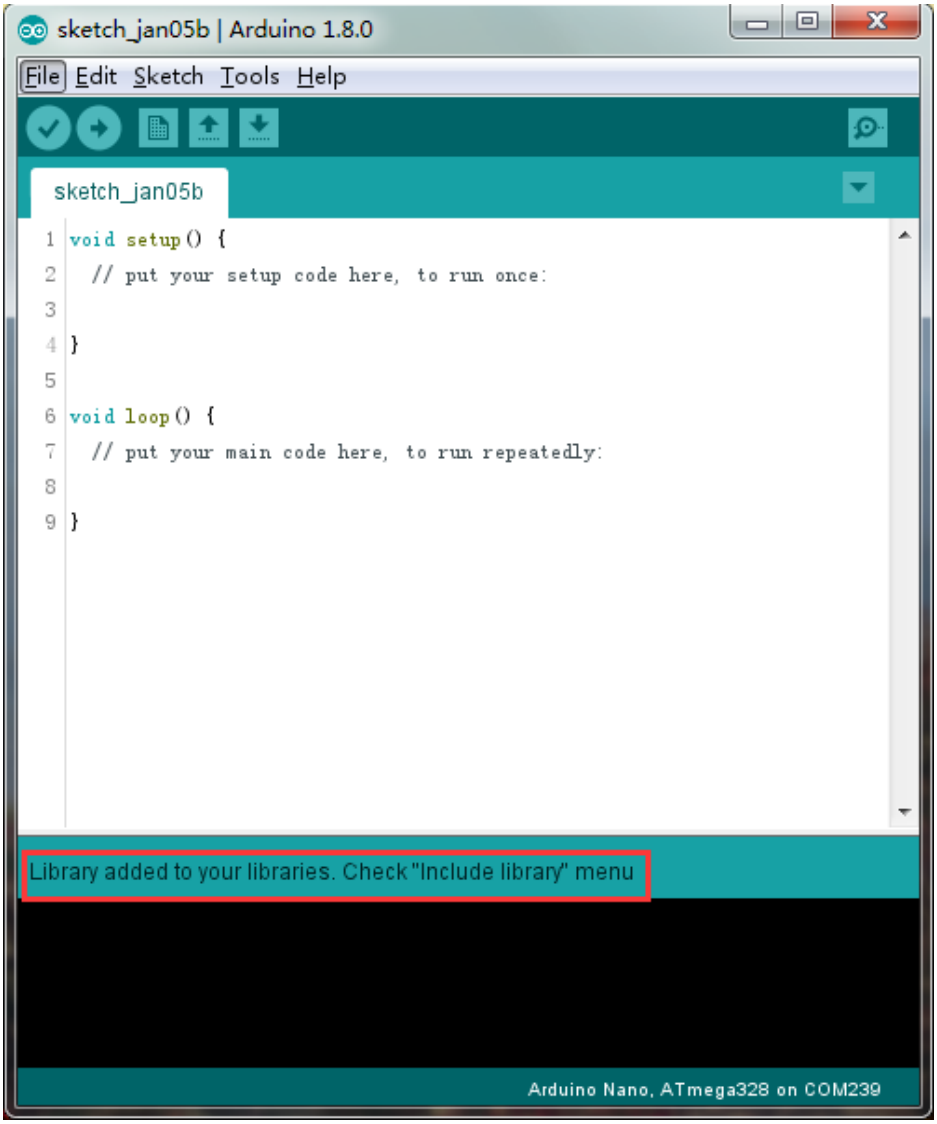
Procurez-vous sur internet (notamment sur GITHUB) le zip de la bibliothèque que vous souhaitez ajouter. Il n'est pas nécessaire de dézipper les fichiers, mais notez bien l'emplacement du fichier.

Dans le menu « Sketch », sélectionner « Inclure une bibliothèque ». Sélectionnez ensuite « Ajouter .ZIP Library ».



Il vous suffit ensuite de sélectionner le zip contenant le fichier





Revenez au menu Sketch> Import Library. Vous devriez maintenant voir la bibliothèque en bas du menu déroulant. Il est prêt à être utilisé dans votre croquis. Le fichier zip sera étendu dans le dossier des bibliothèques dans votre répertoire d'esquisser Arduino. NB: la bibliothèque sera disponible à utiliser dans des croquis, mais les exemples pour la bibliothèque ne seront pas exposés dans le fichier> Exemples jusqu'à la redémarrage de l'IDE.

Ces deux sont les approches les plus courantes. Les systèmes MAC et Linux peuvent être gérés de la même manière. L'installation manuelle à introduire ci-dessous comme alternative peut être rarement utilisée et les utilisateurs sans besoins peuvent l'ignorer.

Installation manuelle

Pour installer la bibliothèque, quittez d'abord l'application Arduino. Ensuite, décompressez le fichier ZIP contenant la bibliothèque. Par exemple, si vous installez une bibliothèque appelée

"ArduinoParty", décompressez ArduinoParty.zip. Il devrait contenir un dossier appeléArduinoParty, avec des fichiers comme ArduinoParty.cpp et ArduinoParty.h à l'intérieur. (Si les fichiers .cpp et .h ne sont pas dans un dossier, vous devrez en créer un. Dans ce cas, vous devez créer un dossier appelé "ArduinoParty" et y déposer tous les fichiers qui se trouvaient dans le ZIP Fichier, comme ArduinoParty.cpp et ArduinoParty.h.)
Faites glisser le dossier ArduinoParty dans ce dossier (votre dossier de bibliothèques). Sous Windows, il sera probablement appelé "Mes documents \ Arduino \ bibliothèques". Pour les utilisateurs de Mac, il sera probablement appelé «Documents / Arduino / bibliothèques». Sur Linux, ce sera le dossier "bibliothèques" dans votre carnet de croquis.

Votre dossier de bibliothèque Arduino devrait maintenant ressembler à ceci (sous Windows):

Mes documents \ Arduino \ bibliothèques \ ArduinoParty \
ArduinoParty.cpp Mes documents \ Arduino \ bibliotecas \ ArduinoParty \
ArduinoParty.h Mes documents \ Arduino \ bibliotecas \ ArduinoParty \
examples

Ou comme ceci (sur Mac et Linux):

Documents / Arduino / bibliothèques / ArduinoParty / ArduinoParty.cpp
Documents / Arduino / bibliothèques / ArduinoParty / ArduinoParty.h
Documents / Arduino / bibliothèques / ArduinoParty / examples.
...

Il peut y avoir plus de fichiers que les fichiers .cpp et .h, assurez-vous qu'ils sont tous là. (La bibliothèque ne fonctionnera pas si vous mettez les fichiers .cpp et .h directement dans le dossier des bibliothèques ou si elles sont imbriquées dans un dossier supplémentaire. Par exemple: Documents \ Arduino \ bibliotecas \ ArduinoParty.cpp et Documents \ Arduino \ Les bibliothèques \ ArduinoParty \ ArduinoParty \ ArduinoParty.cpp ne fonctionneront pas.)

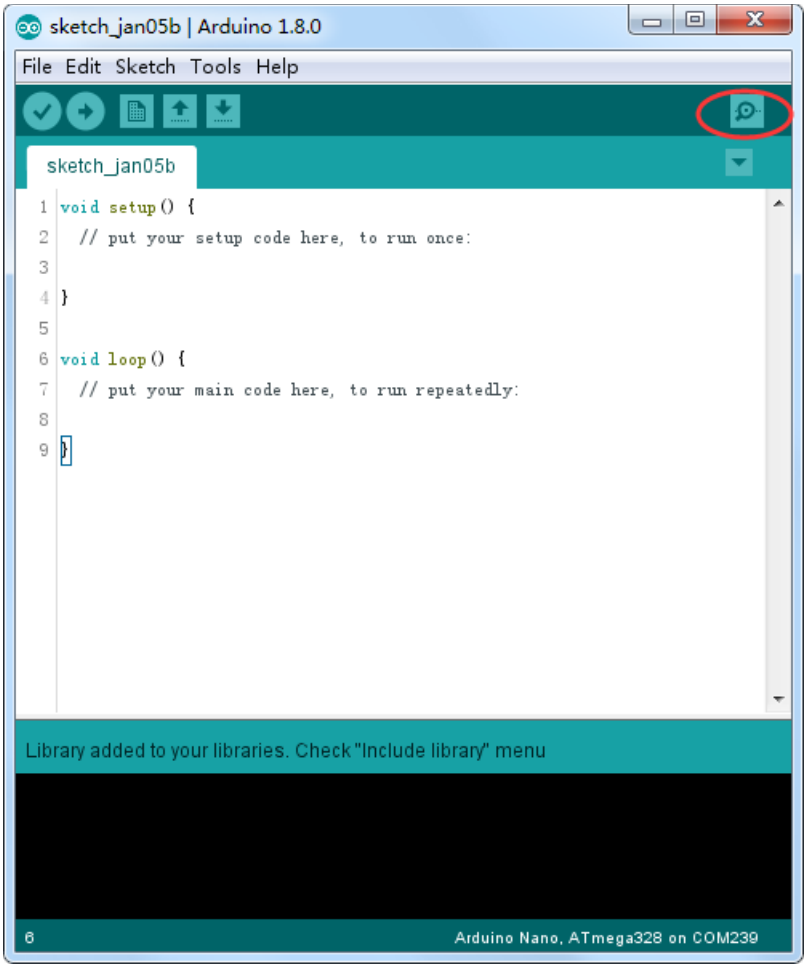
Redémarrez l'application Arduino. Assurez-vous que la nouvelle bibliothèque apparaît dans l'élément de menu Sketch-> Import Library du logiciel. C'est tout! Vous avez installé une bibliothèque!

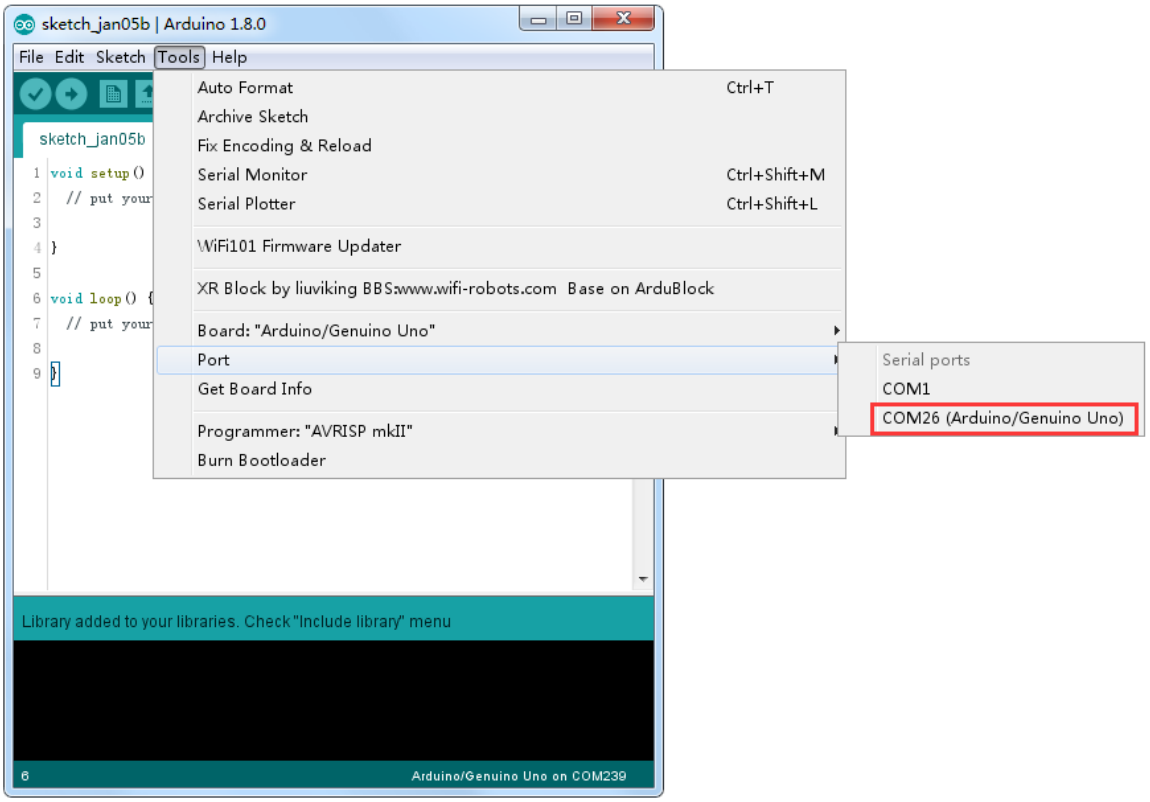
Utilisation du moniteur série (Windows, Mac, Linux)

Le moniteur série est une petite fenêtre très utile qui va vous permettre d'interagir en temps réel avec la carte UNO. Vous pouvez lui envoyer des informations et elle pourra à son tour faire de même. Vous verrez dans les différentes leçons que le recours au moniteur série est très fréquent.

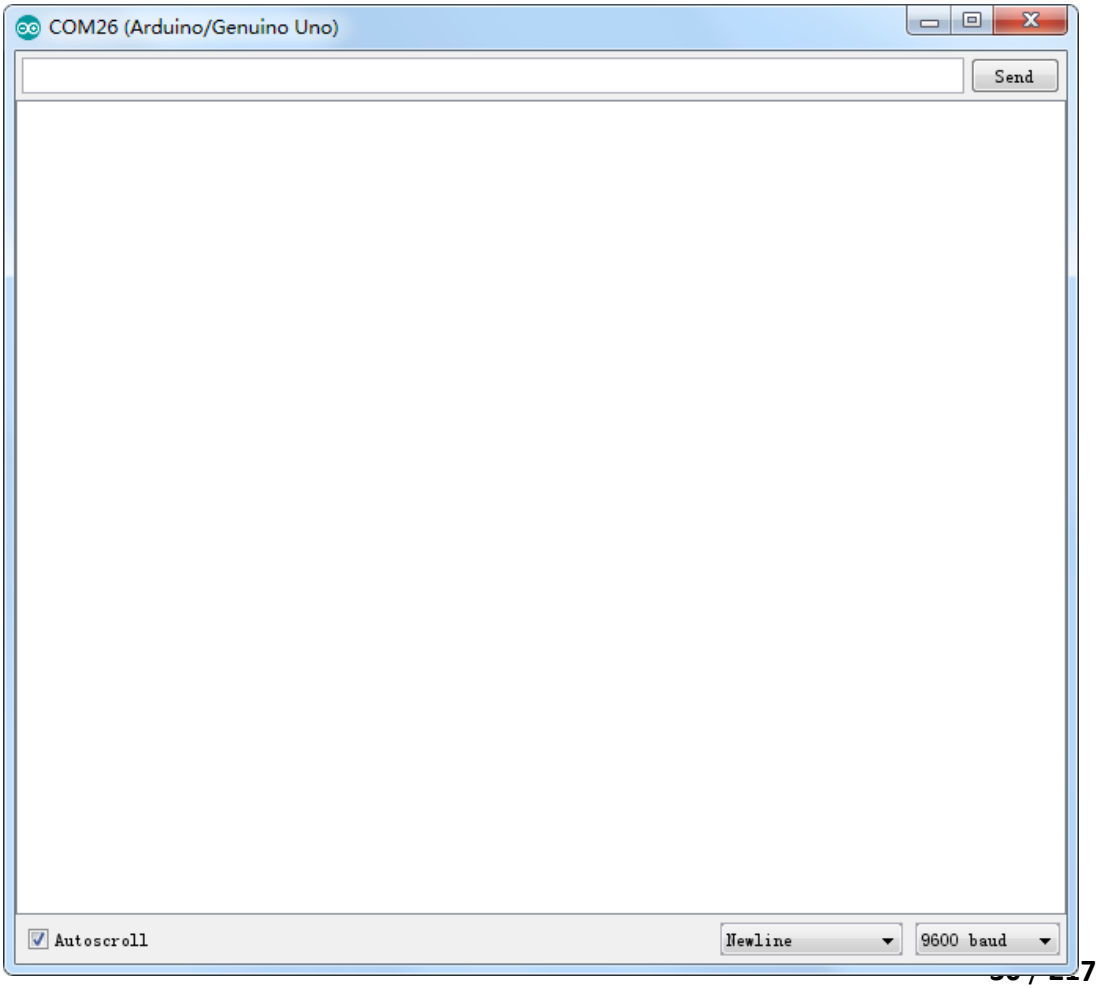
Se Connecter

Pour l'ouvrir, c'est très simple, il suffit de cliquer sur le petit icône entouré en rouge.



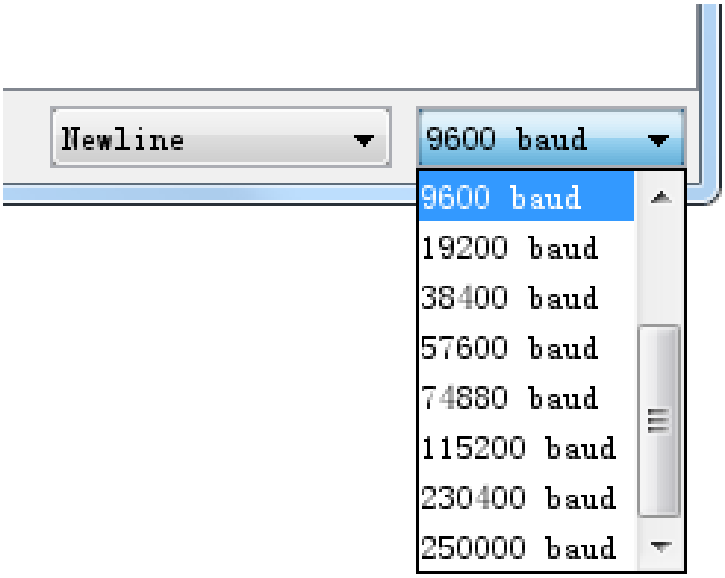


A l'ouverture, vous devez voir cette fenêtre:

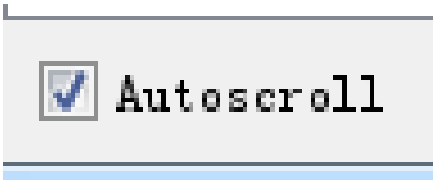


Settings

Vous pouvez définir le taux de transfert de la connexion de la manière suivante



Et vous pouvez avoir un défilement automatique de l'écran comme suit



Avantages

Le Serial Monitor est un excellent moyen d'établir rapidement une connexion série avec votre Arduino. Si vous travaillez déjà dans l'IDE Arduino, il n'est vraiment pas nécessaire d'ouvrir un terminal distinct pour afficher les données.

Les inconvénients

Le manque de paramètres laisse beaucoup à désirer dans le Serial Monitor, et, pour les communications en série avancées, il se peut que ce ne soit pas le tour.

Leçon 2 Blink

But de la leçon

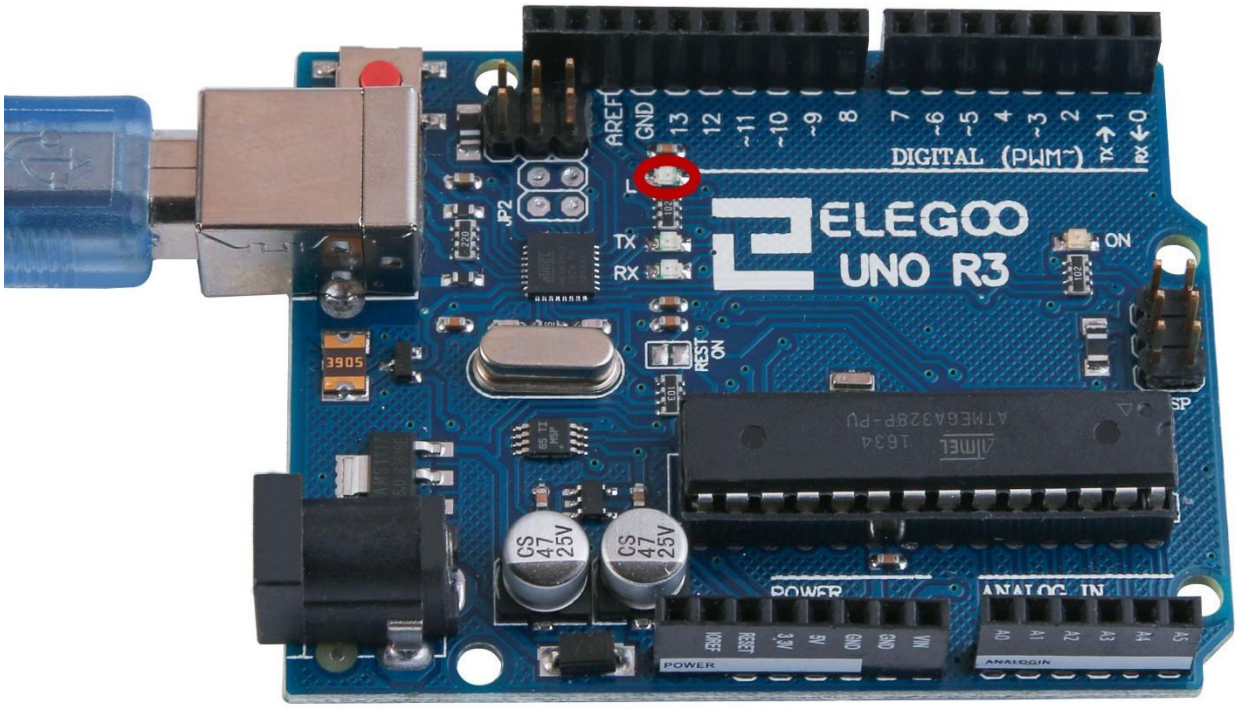
Dans cette première leçon, vous allez prendre en main l'IDE et apprendre à faire clignoter la LED intégrée à la carte UNO R3.

Matériel nécessaire:

(1) x Elegoo Uno R3

Principe

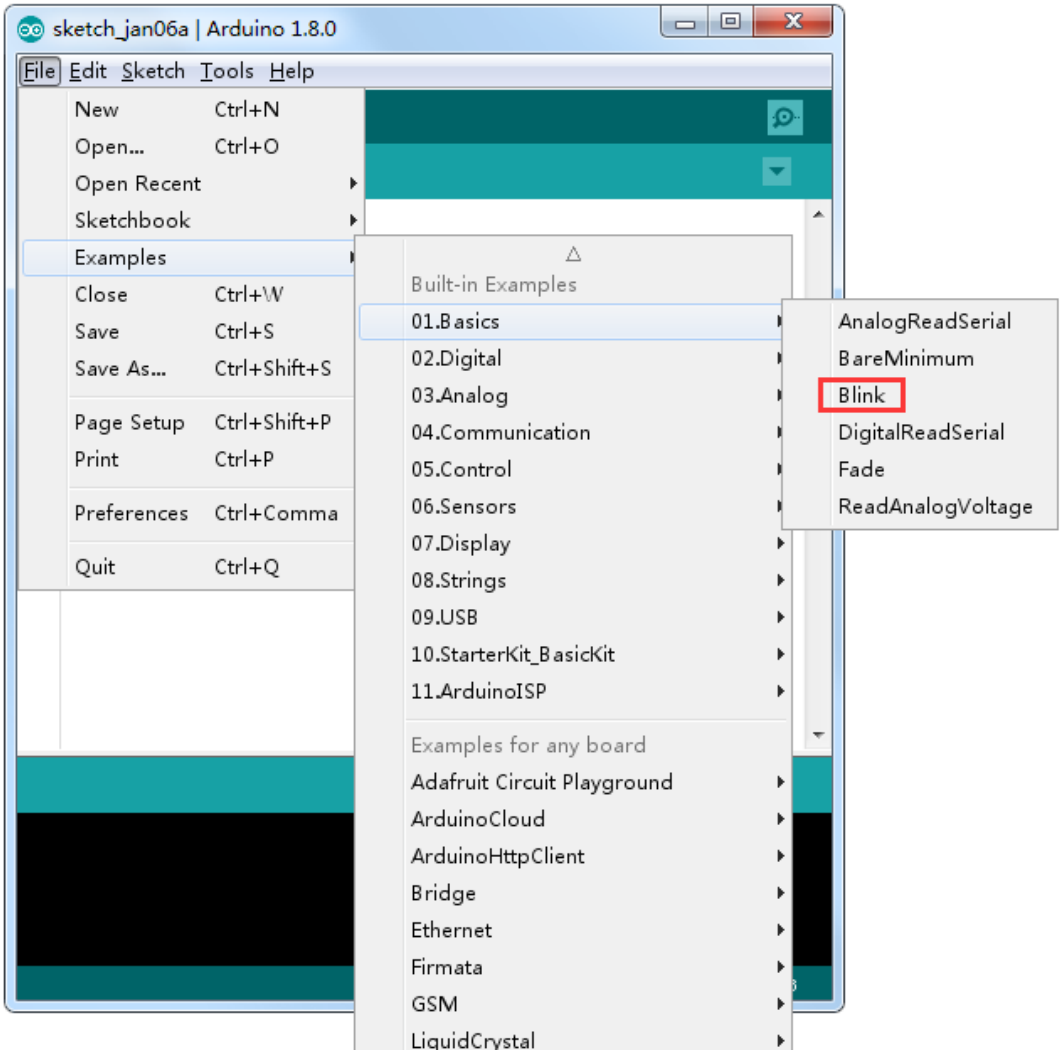
La carte UNO R3 possède deux rangées de connecteurs le long de ses extrémités qui sont utilisés pour brancher une vaste gamme de composants électroniques ou des cartes d'extensions (appelées shields) qui augmentent ses capacités.
Elle est aussi équipée d'une LED intégrée qu'il est possible de commander au travers de vos programmes. Vous pouvez apercevoir cette LED sur l'image ci-dessous, elle est repérable grâce au « L » visible sur la carte.



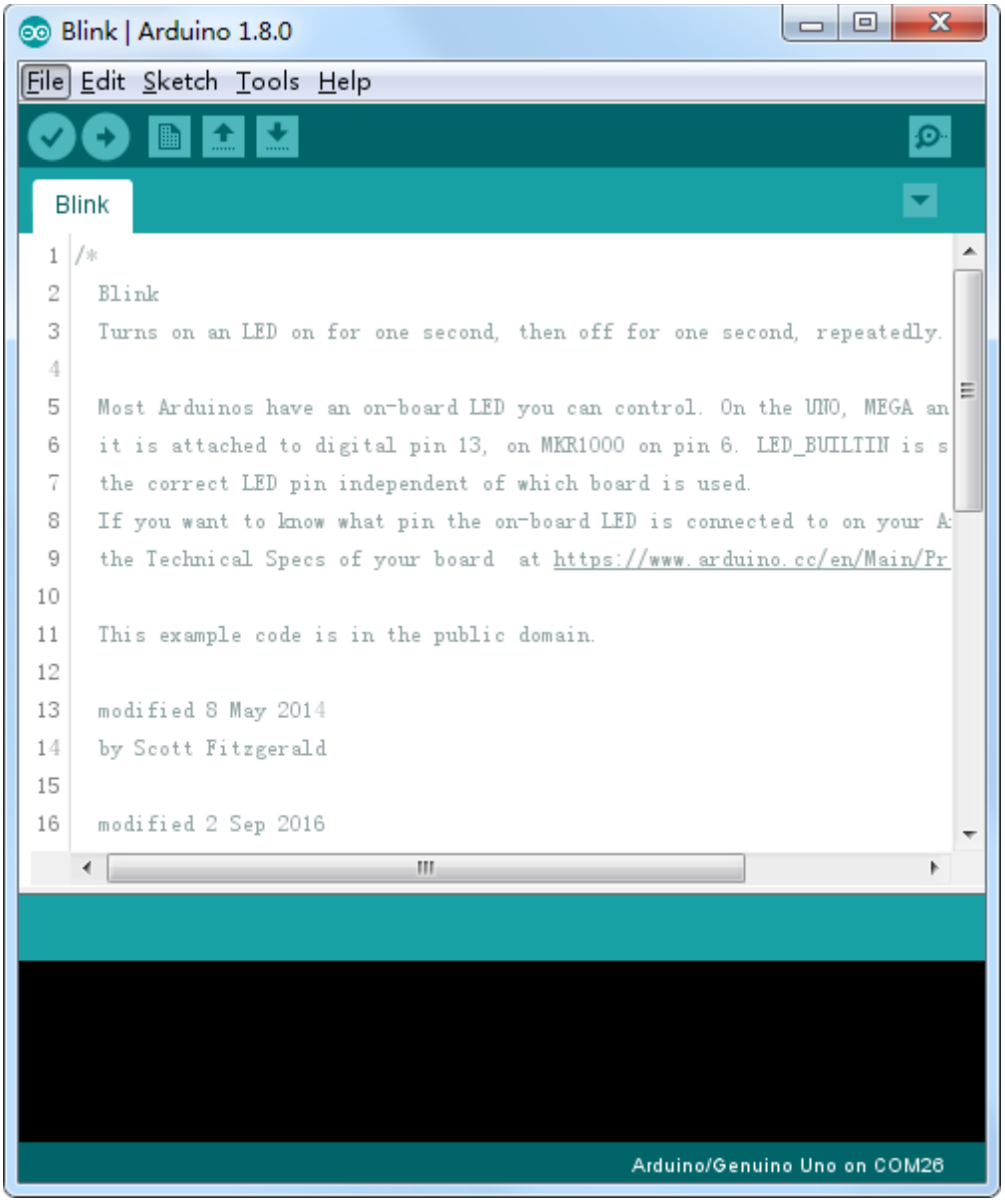
Lorsque vous connectez votre carte pour la première fois, il est possible que la LED se mette à clignoter. C'est parce que les cartes sont fréquemment expédiées avec le programme « BLINK » préinstallé.

Dans cette leçon, nous allons reprogrammer la carte UNO R3 avec notre propre programme « BLINK », ce qui va nous permettre notamment de changer la fréquence de clignotement de la LED.

Ouvrez le sketch "BLINK" dans l'environnement de programmation via le menu "FICHIER/EXEMPLES/01.BASICS/Blink"

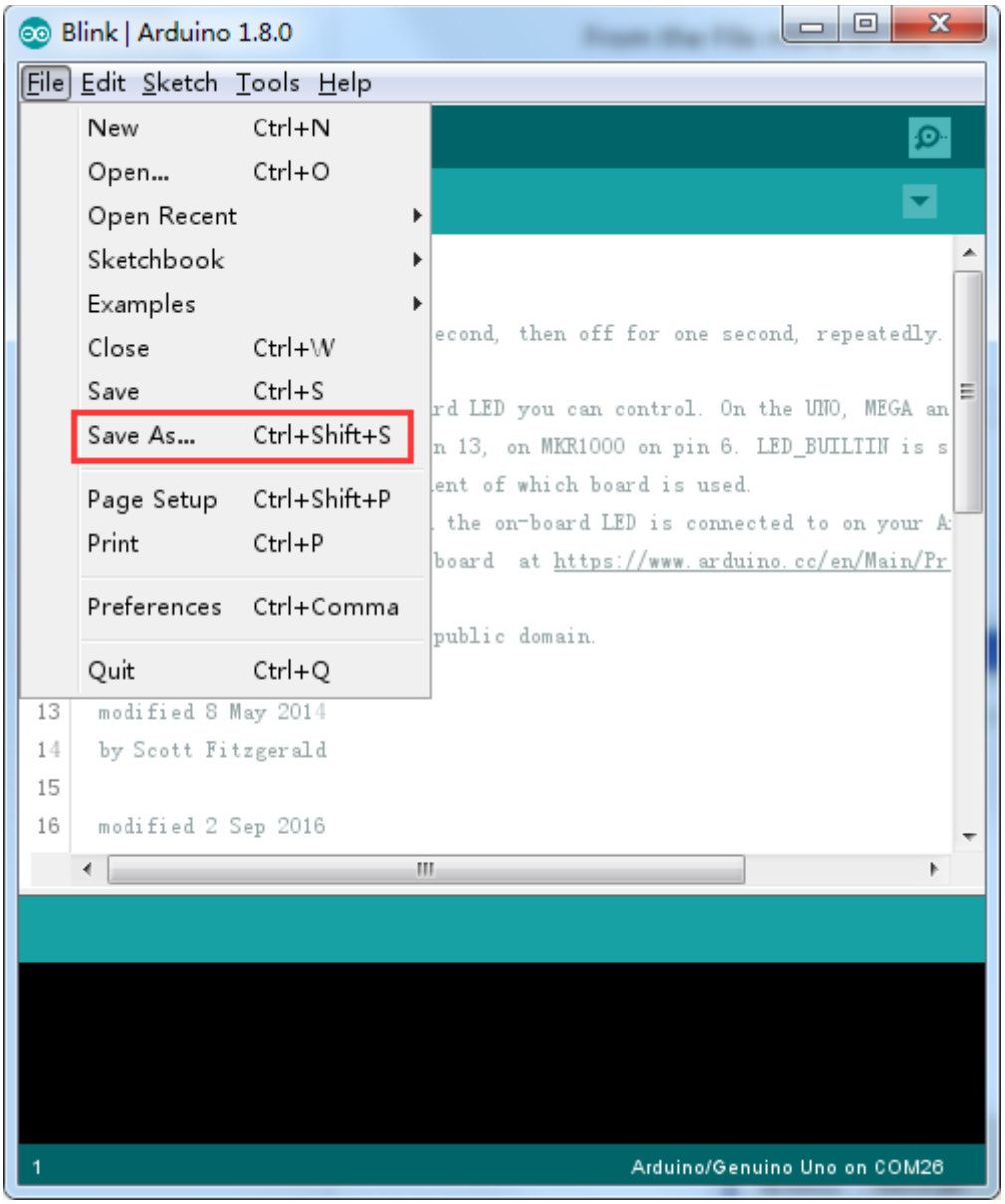


Voici ce que vous devez obtenir.

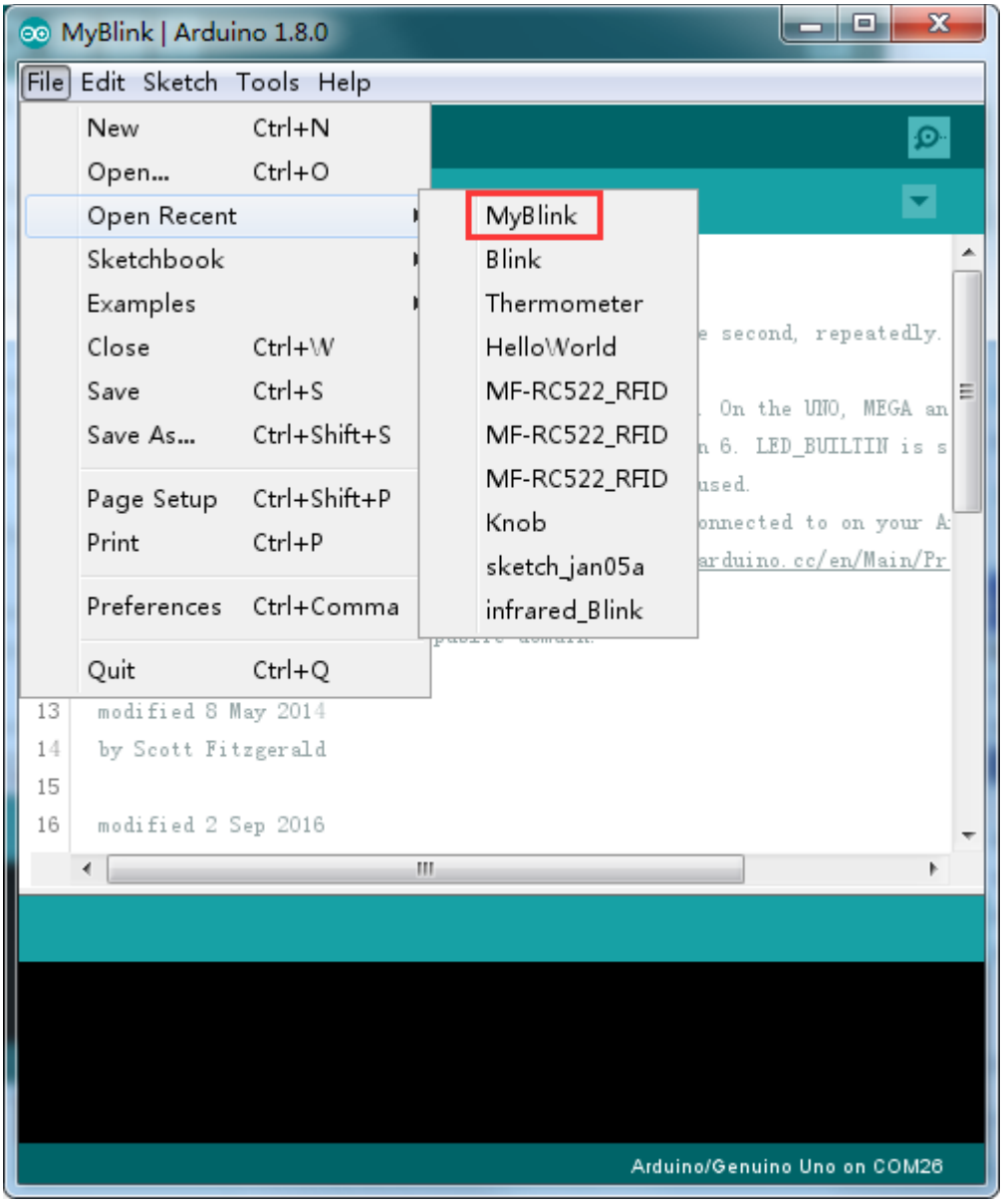


Comme vous allez apporter des modifications à ce sketch, la première étape consiste à l'enregistrer dans votre répertoire personnel, les sketchs exemples étant en lecture seule.

Utilisez le menu « enregistrer sous… » et définissez le répertoire et nom de fichier que vous souhaitez.

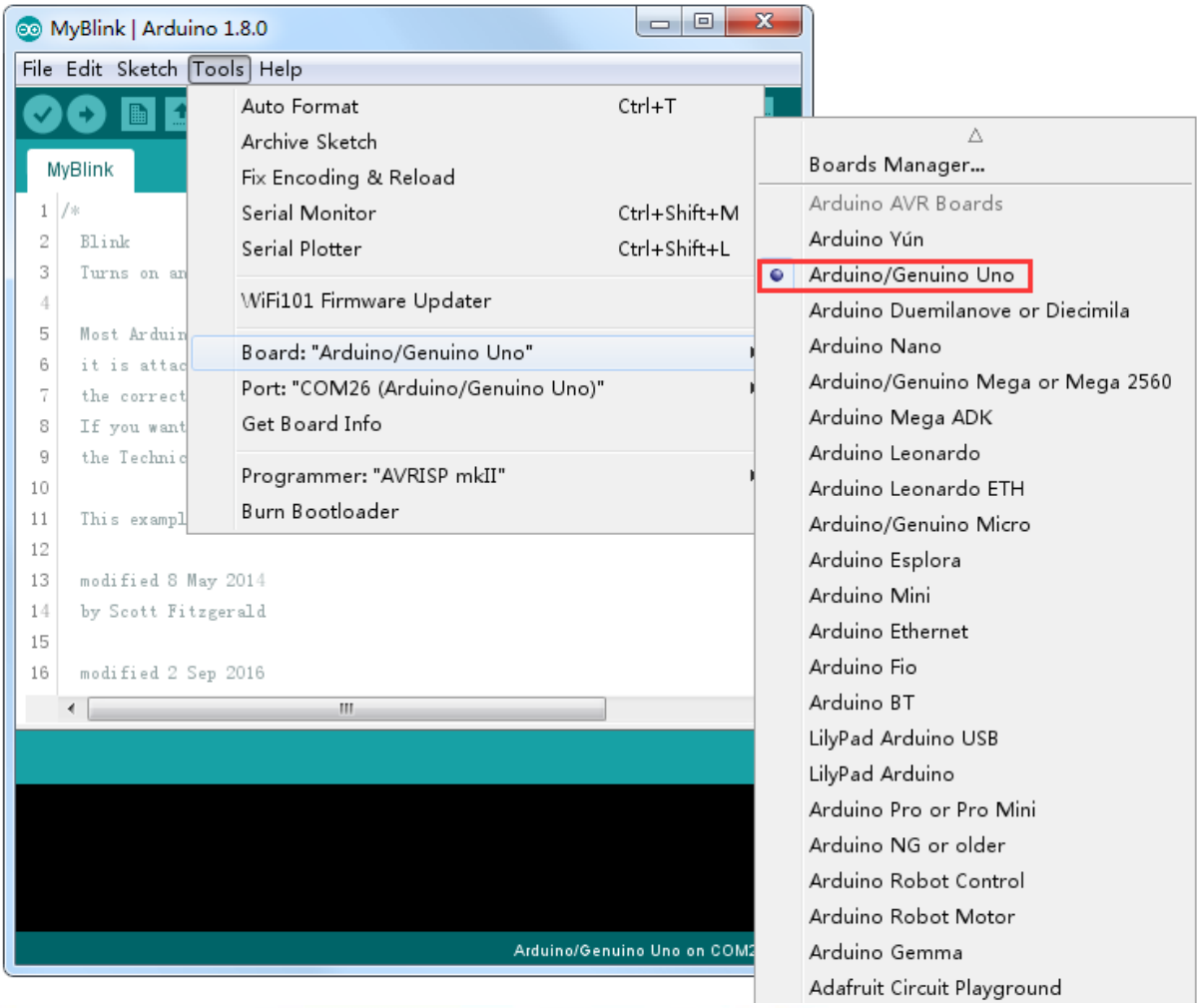


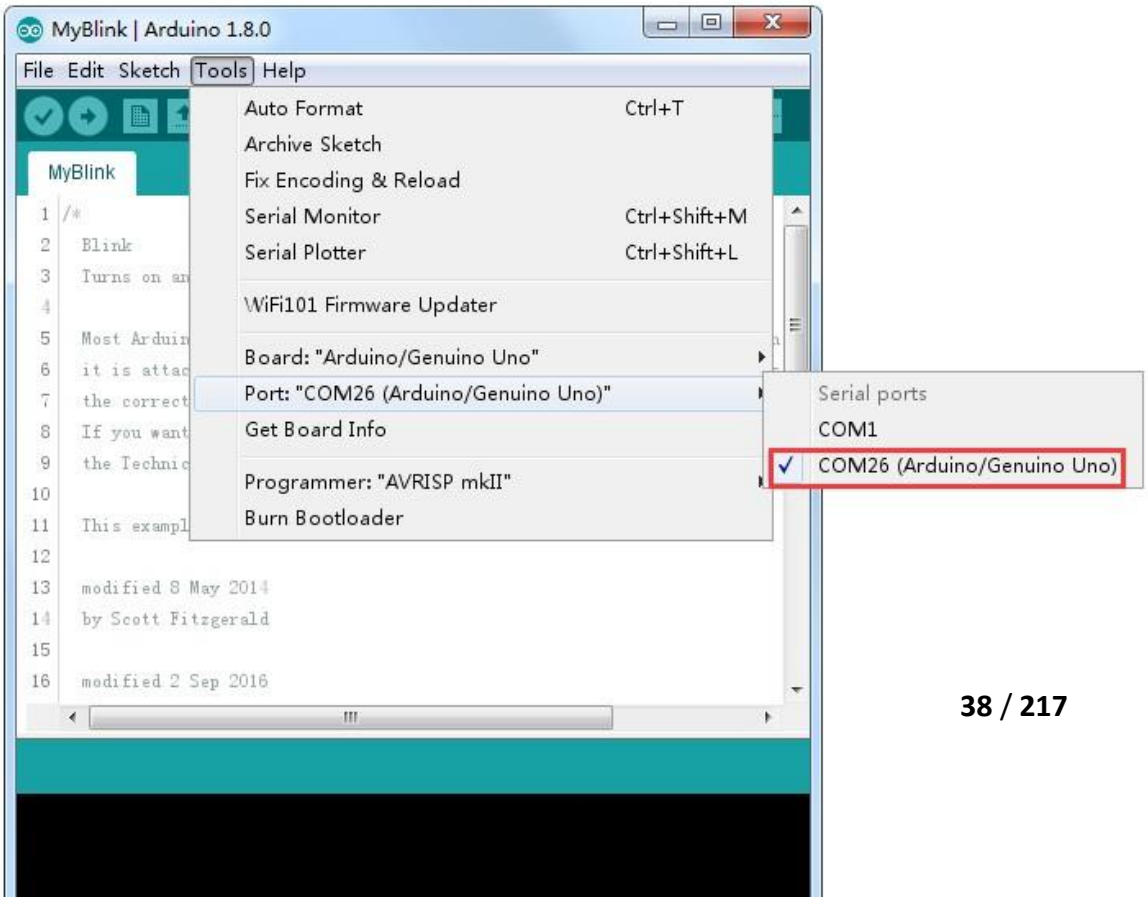
Vous venez d’enregistrer le fichier dans votre répertoire.



Vous pouvez accéder à vos fichiers récents de la même manière que la plupart de vos autres logiciels.

Il est temps maintenant de connecter la carte UNO R3 via le port USB de votre ordinateur et de vérifier la bonne reconnaissance de celle-ci.

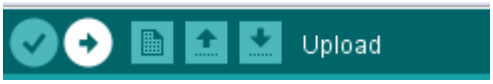




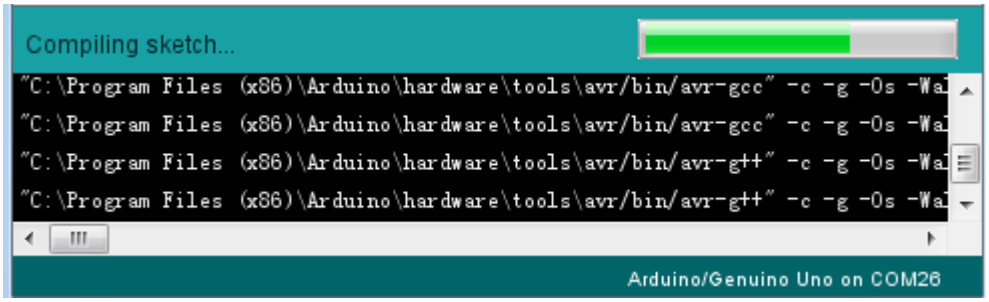
Note: le numéro de port COM ne sera pas nécessairement celui que vous pouvez apercevoir sur les captures d'écran. En effet, c'est votre ordinateur qui va affecter une référence au moment de la connexion et si vous possédez plusieurs cartes, chacune aura potentiellement un numéro différent.

Une fois connecté, vous pouvez apercevoir ce petit texte en bas à droite de votre écran.

Cliquez sur le bouton avec la flèche pour déclencher le téléversement.



Vous pouvez constater que la première étape est la compilation (le logiciel vérifie l'intégrité du code).



Ensuite, le téléversement à proprement parler commence.

Enfin, le statut "terminé" s'affiche.

Un message vous renseigne sur la taille du programme et la quantité de mémoire utilisée sur la carte. Si votre carte n'est pas correctement connectée ou reconnue, vous obtiendrez l'écran suivant. Reportez-vous aux explications données dans les premières leçons pour veillez à avoir correctement installé les drivers.

Vous pouvez constater qu'une part importante du code est consacrée aux commentaires. Les commentaires sont importants ils permettent de bien comprendre le code, surtout quand celui-ci est complexe. Cela permet d'y revenir plus tard lorsqu'il est nécessaire de le faire évoluer.

Il existe deux façons de mettre du texte en commentaire. La première est de commencer la ligne par « // ». La deuxième manière permet de placer une portion de code/texte en commentaire (idéal lorsque l'on veut tester plusieurs alternatives par exemple). Débuter la zone commentaire par « /* » et terminez celle-ci par « */ »

Dans le code « blink », la vraie première ligne d'instruction est la suivante:

```
int led = 13;
```

Cette instruction permet de créer une constant nommée "led" et de lui affecter la valeur numérique "13".

Le bloc suivant représente le code qui sera exécuté une unique fois lors de la mise sous tension de la carte.

```
void setup() {
// initialize the digital pin as an output.
pinMode(led, OUTPUT);
}
```

Tous les sketchs doivent obligatoirement avoir une bloc “setup” dans lequel vous pouvez placez autant d’instructions que nécessaires. Les instructions s’écrivent entre {}.

Dans « BLINK » la seule instruction consiste à affecter le pin 13 en tant que sortie (OUTPUT).

Il est aussi obligatoire dans un sketch d’avoir un bloc “loop”. Contrairement au premier bloc celui-ci s’exécutera en boucle tant que la carte sera sous tension.

```
void loop() {
   digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
   delay(1000);                        // wait for a second
   digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
   delay(1000);                        // wait for a second
}
```

Le bloc « loop » est constitué de 4 instructions. La première déclenche l’allumage de la led, la deuxième une attente de 1000ms, la troisième l’extinction de la led. La dernière instruction une attente de 1000ms.

Vous allez maintenant augmenter la fréquence de clignotement de la LED. Vous avez certainement deviné que le paramètre à changer est celui entre () sur les instructions « delay ». Téléversez le programme et observez.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

Leçon 3 LED

But de la leçon

Dans celle leçon, vous allez apprendre à faire varier la luminosité d'une LED en utilisant plusieurs valeurs de résistances.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x LED rouge de 5mm

(1) x résistance 220 ohm

(1) x résistance 1 kohm

(1) x résistance 10 kohm

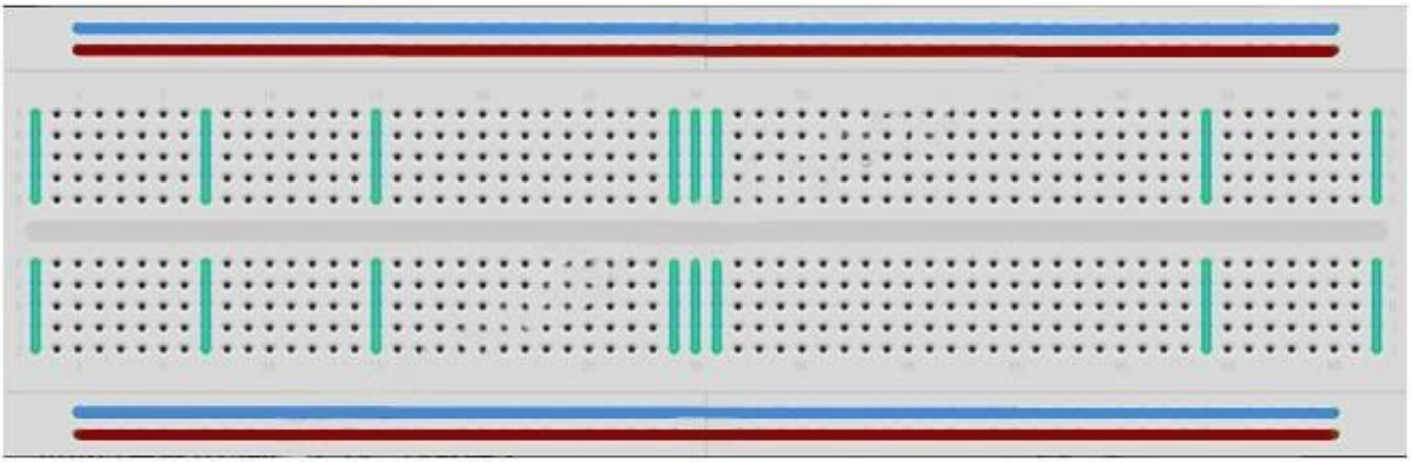
(2) x câbles mâle-mâle

Présentation des composants

PLANCHE PROTOTYPE MB-102：

Une planche prototype vous permet de réaliser des circuits très rapidement, sans avoir besoin de réaliser de soudures.

Exemple :

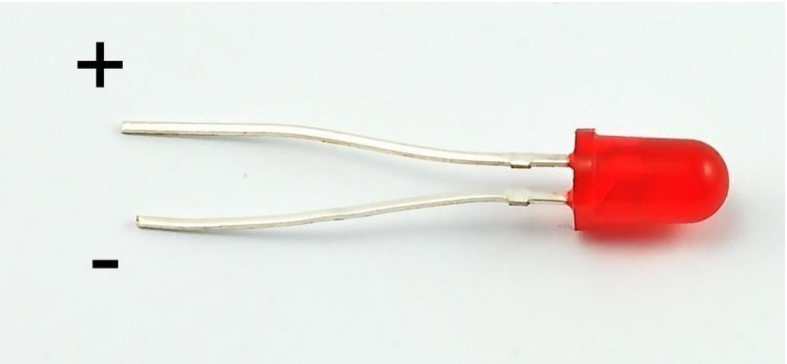


Il existe une grande variété de planches prototypes. La plus simple est une grille de trous dans un bloc de plastique. A l'intérieur se trouvent des lames métal permettant la connexion électrique entre les différents trous d'une même ligne. Mettre les branches de deux composants sur une même ligne les "assemble" électriquement. La ligne creusée au centre de la plaque symbolise une rupture électrique entre la partie haute et la partie basse. Cela signifie aussi que vous pouvez positionner une puce à cheval entre haut et bas. Certaines planches prototypes ont aussi deux lignes horizontales en haut et en bas. On les utilise généralement pour créer une ligne d'alimentation +5V et masse.
Attention tout de même, les planches prototypes ont comme limite d'utilisation la qualité des connexions qui ne valent pas une soudure et peuvent entraîner parfois des dysfonctionnements.

LED:

Les leds font de parfaits indicateurs lumineux. Elles consomment peut de courant et ont une très bonne durée de vie.
Dans cette leçon, vous allez utiliser certainement le type de leds le plus commun, la led 5mm. Il en existe d'autres de 3 à 10mm.

Attention: il n'est pas possible de brancher directement une led sur une batterie car l'intensité de courant est trop forte.



Attention aussi au sens de branchement de la led. La patte la plus longue doit être connectée à la borne + et la patte courte à la borne -.

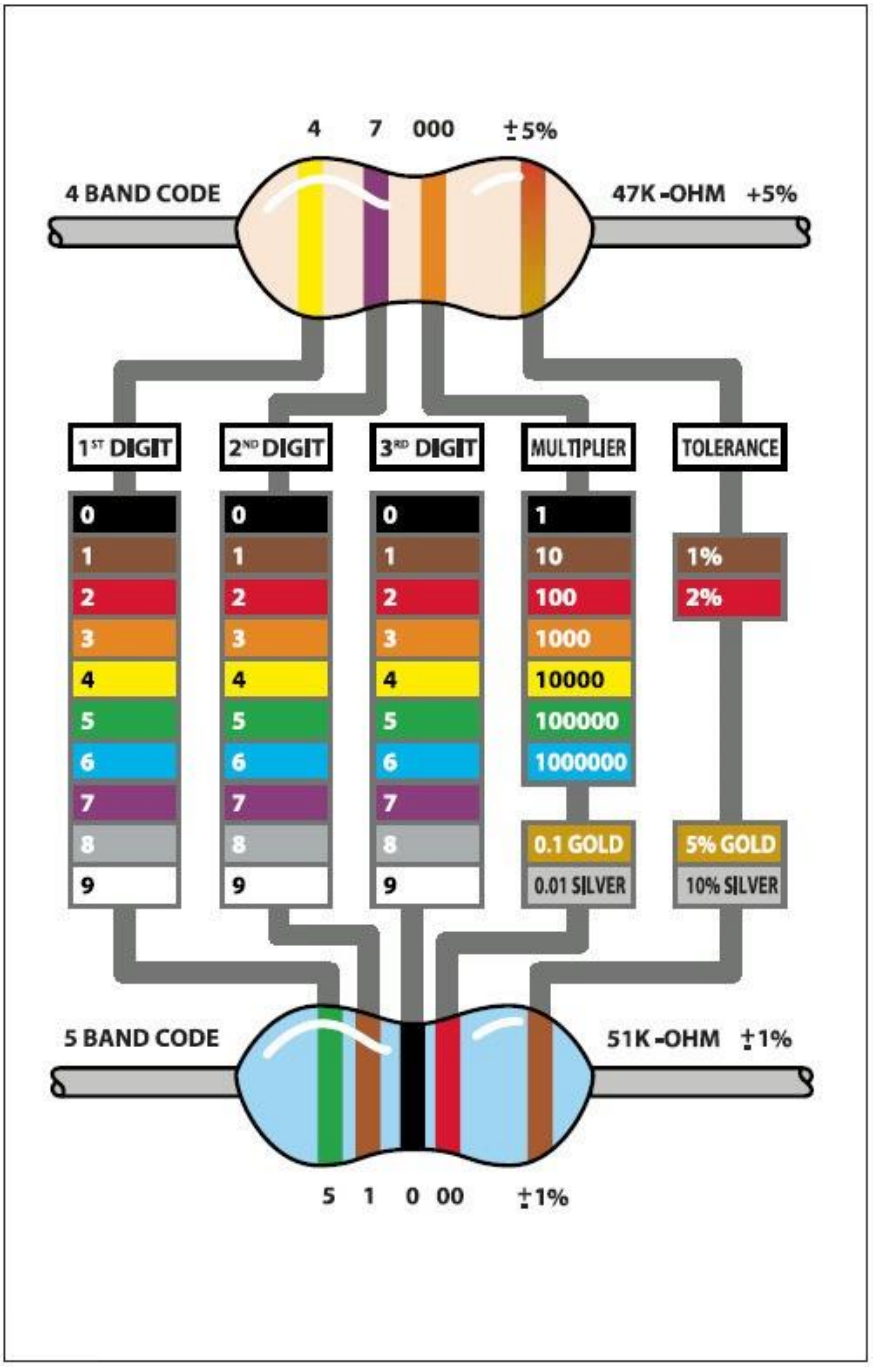
RESISTANCES:

Comme leur nom le suggère, les résistances s'opposent au passage du flux d'électricité. Plus la valeur est grande plus la résistance l'est aussi. Vous allez pouvoir calibrer la brillance de la led en jouant sur la valeur de la résistance.

Mais tout d'abord, quelques informations supplémentaires…

L'unité des résistances est l'Ohm, que l'on représente généralement avec la lettre grecque Ω. Il faut beaucoup d'ohms pour avoir un minimum de résistance, c'est pour cela que l'on retrouve généralement des kilos, voir mégas ohms : kΩ (1,000 Ω) et MΩ (1,000,000 Ω).

Dans cette leçon, vous allez utiliser trois valeurs de résistance : 220Ω, 1kΩ and 10kΩ. C'est grâce aux anneaux de couleurs et au code associé que l'on peut connaitre la valeur d'une résistance.



Notez aussi que contrairement aux leds, les résistances n'ont pas de sens.

Pour simplifier la tâche, il est aussi possible d'utiliser un appareil de mesure afin de connaître la valeur de la résistance.

Connexion

Schéma de câblage

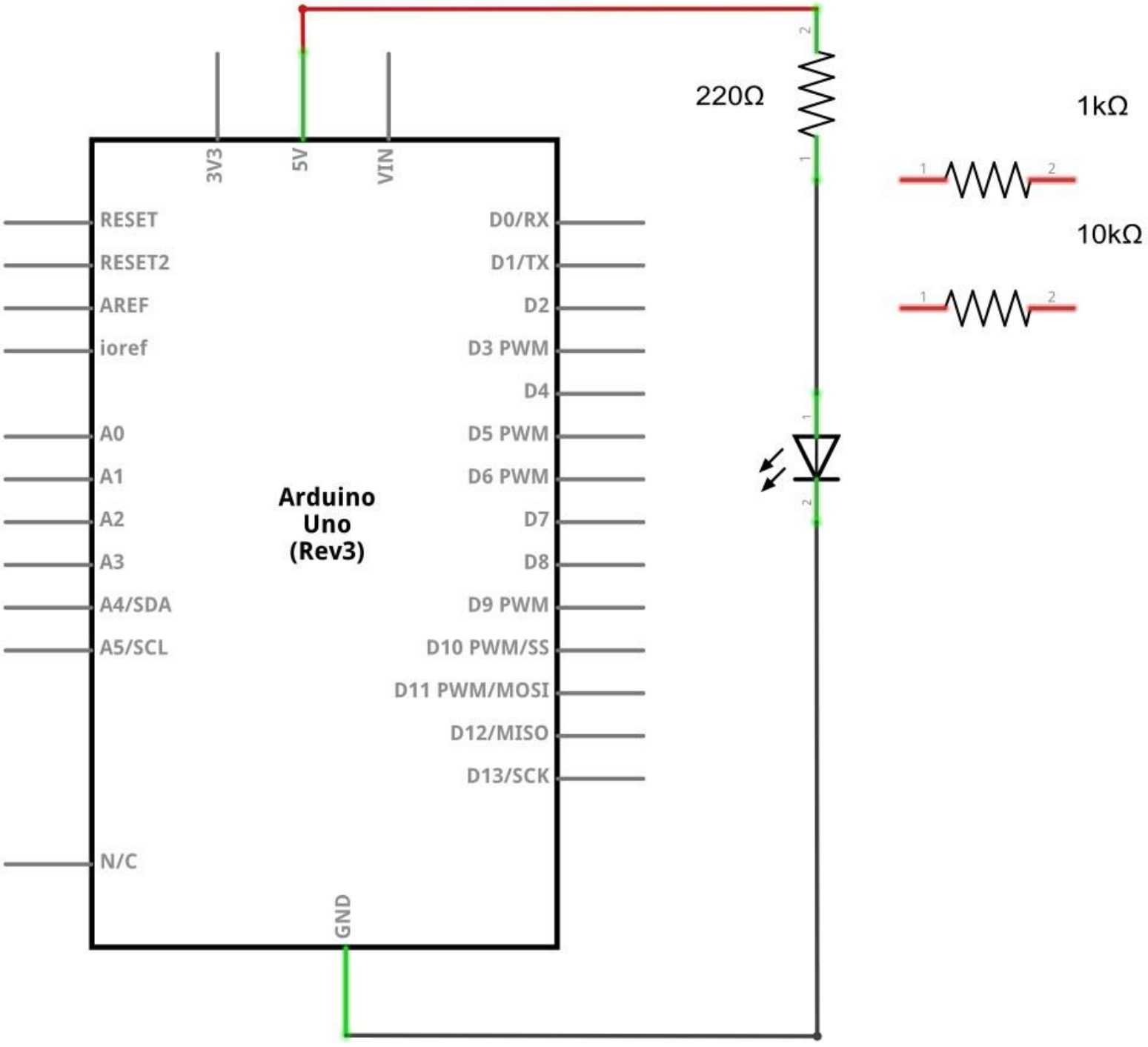
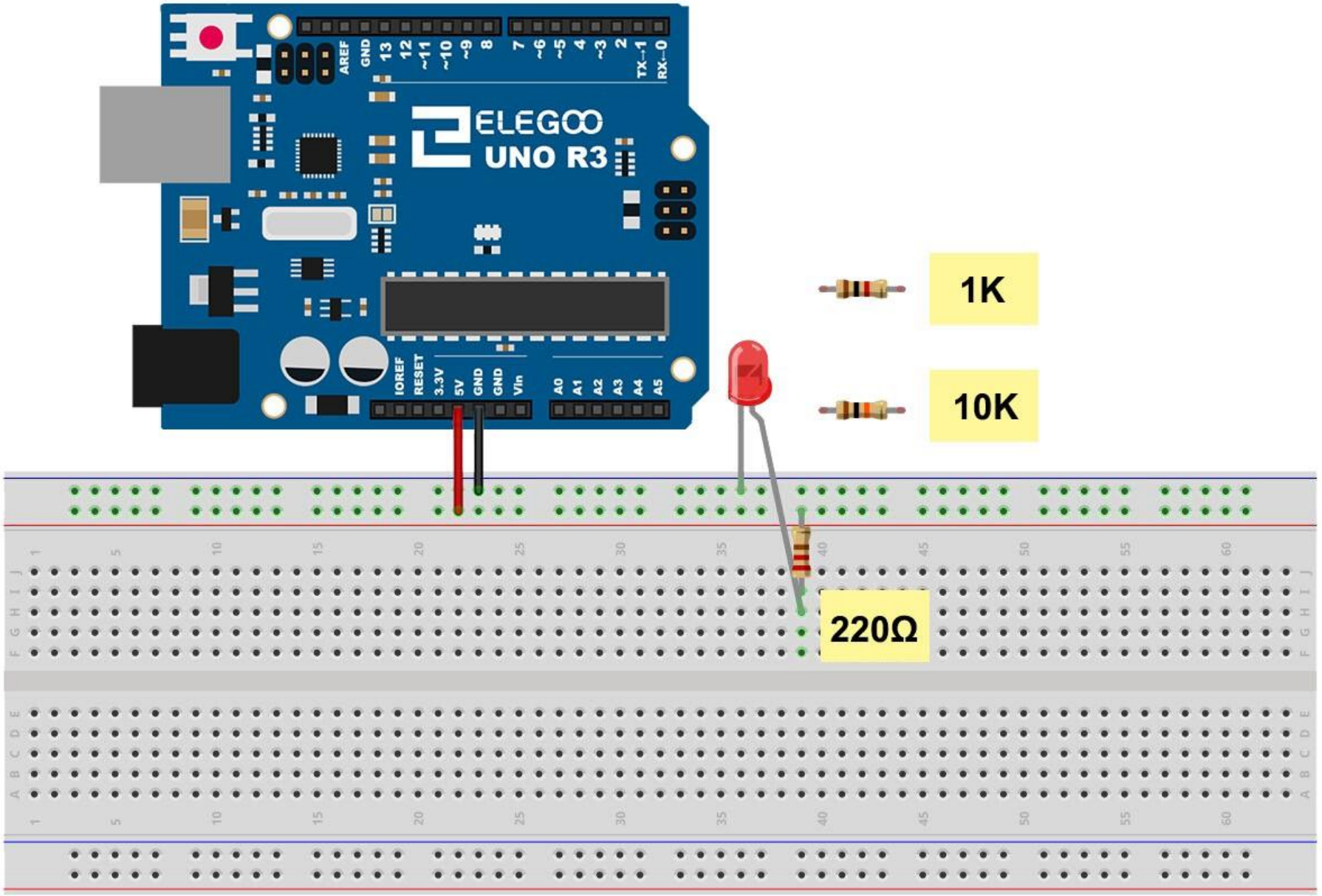


Diagramme de câblage

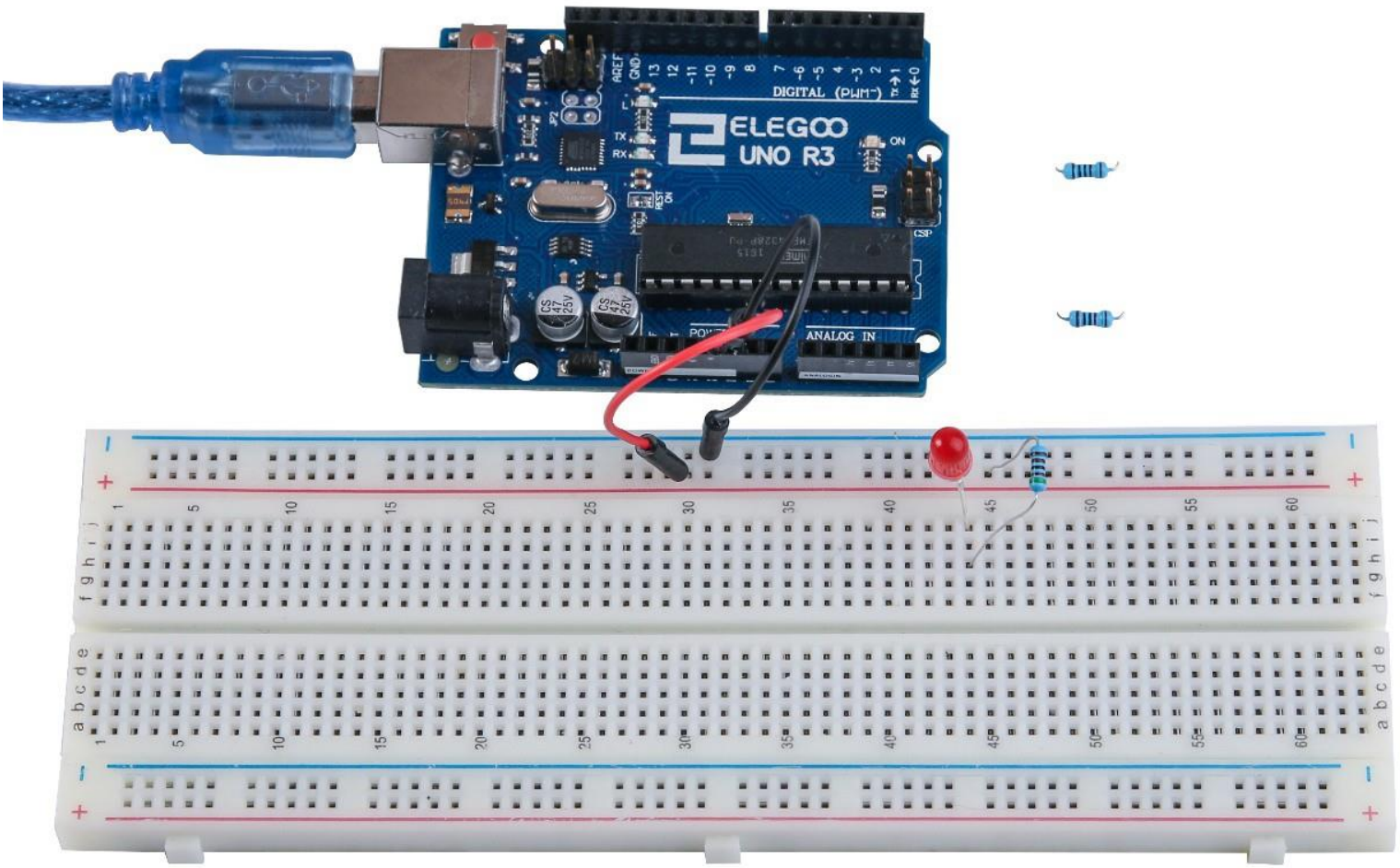


L’UNO est une bonne source de +5V. Vous allez donc pouvoir l’utiliser pour fournir le bon voltage à la LED et la résistance. Il n’est pas nécessaire de téléverser un code pour cet exemple, simplement connecter la carte à un ordinateur pour celle-ci soit sous tension.

Avec une résistance de 220 Ω, la LED est très brillante. Elle l’est un peu moins avec une résistance de 1kΩ et à peine brillante avec une résistance de 10 kΩ.

Vous pouvez positionner la résistance avant ou après la led et constater que cela n’a pas d’importance.

Illustration



Leçon 4 RGB LED

But de la leçon

Les LED RGB sont une façon simple et amusante d'ajouter de la couleur à vos projets.
Il en existe deux types, à anode commune et à cathode commune.
A anode commune utilisent le +5V en entrée commune, à cathode commune la masse en sortie commune.
Il ne faut pas oublier d'utiliser des résistances pour limiter le courant et protéger les leds de la destruction.
Dans le projet, vous allez commencer avec une couleur Verte, passer par le Bleu pour enfin terminer avec le Rouge.

Matériel nécessaire:

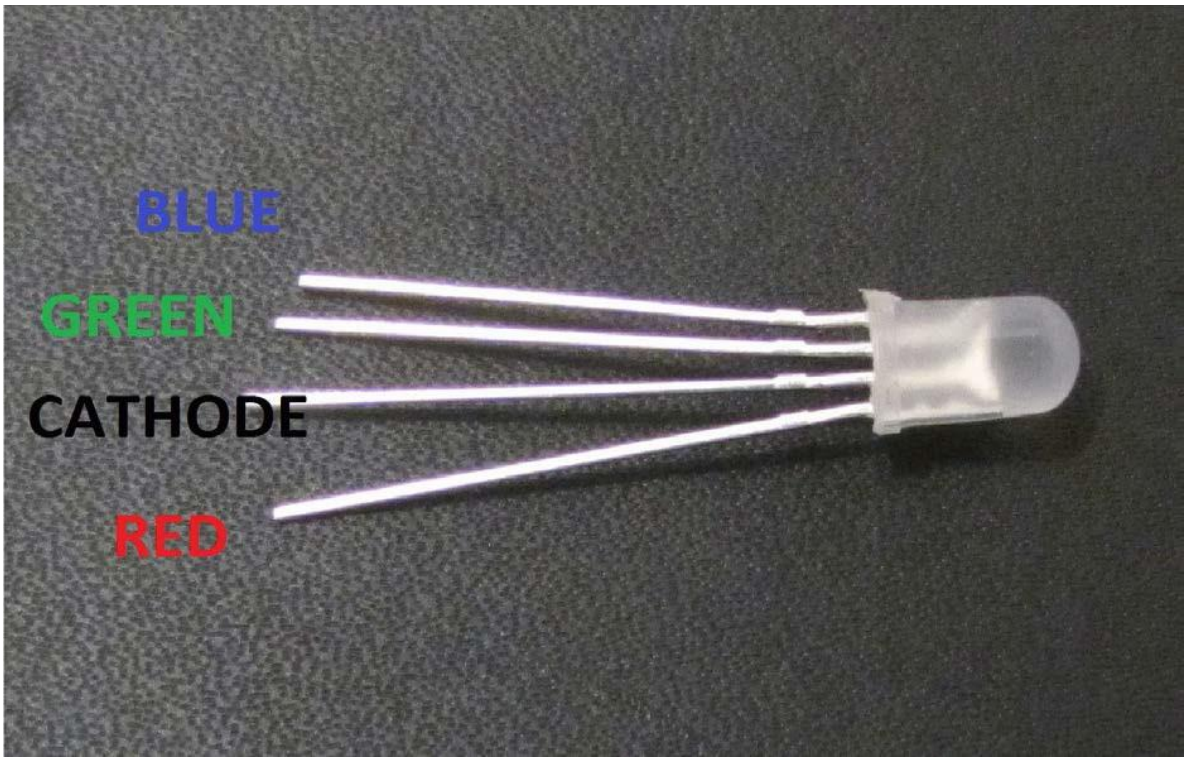
(1) x Elegoo Uno R3
(1) x Planche prototype
(4) x Câbles mâle-mâle
(1) x led RGB
(3) x résistance 220 ohms

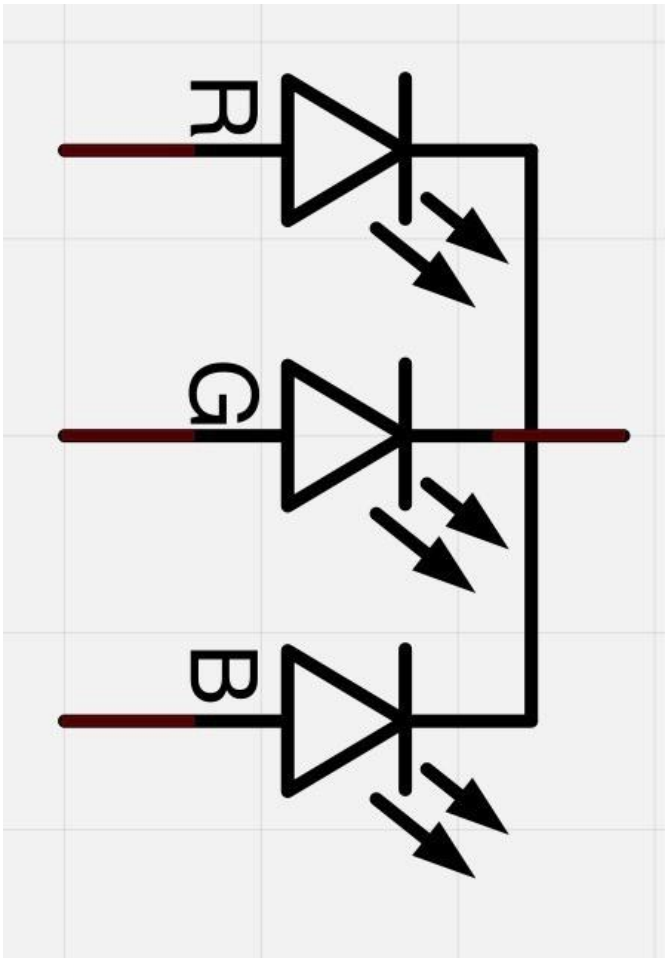
Présentation des composants

RGB:

Au premier abord, la led ressemble à une led standard monochrome. En fait, à l'intérieur, ce sont 3 leds qui sont installées : une bleue, une rouge et une verte. Il est possible en mixant les couleurs de créer une large palette de coloris. Pour cela, il suffit d'ajuster la brillance de chaque led. On pourrait pour cela utiliser des valeurs de résistances différentes, un travail bien compliqué pour obtenir la couleur souhaitée. Heureusement, la carte UNO R3 va être très utile. Grâce à ses sorties analogiques, l'ajustement d'intensité sera un jeu d'enfant.

L'illustration suivante montre comment se répartissent les différentes pattes de la led.



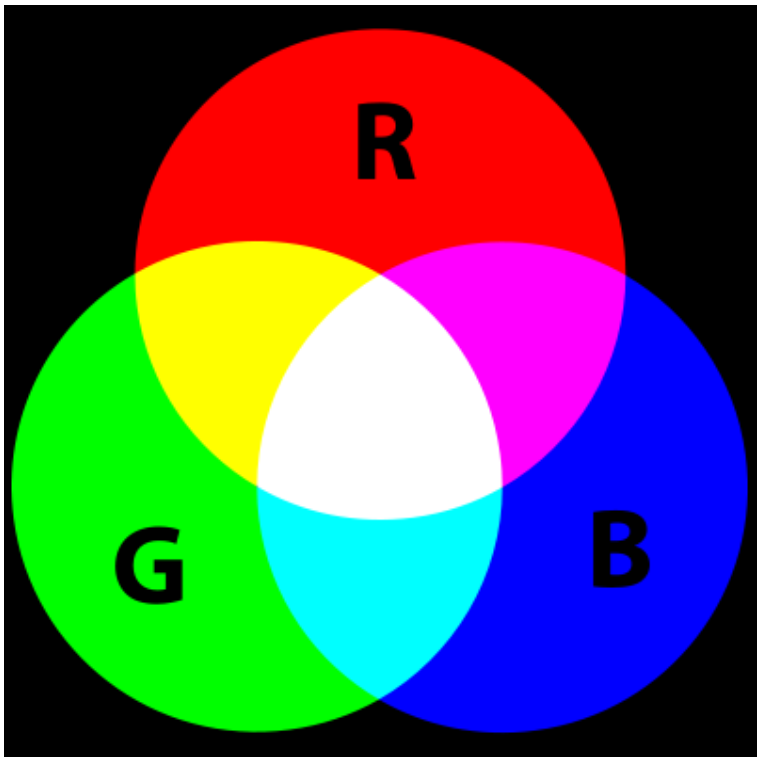


Il faut connecter une sortie analogique de la carte UNO R3 sur les pattes ROUGE/VERT/BLEU et une masse sur la patte (GND)

N’oubliez pas encore une fois de positionner une résistance 220 ohms avant chaque patte d’entrée.

COLEURS:

L'illustration suivante montre comment se créent les différentes couleurs en fonction de l'association des couleurs de bases. Servez-vous en pour créer vos couleurs personnalisées.

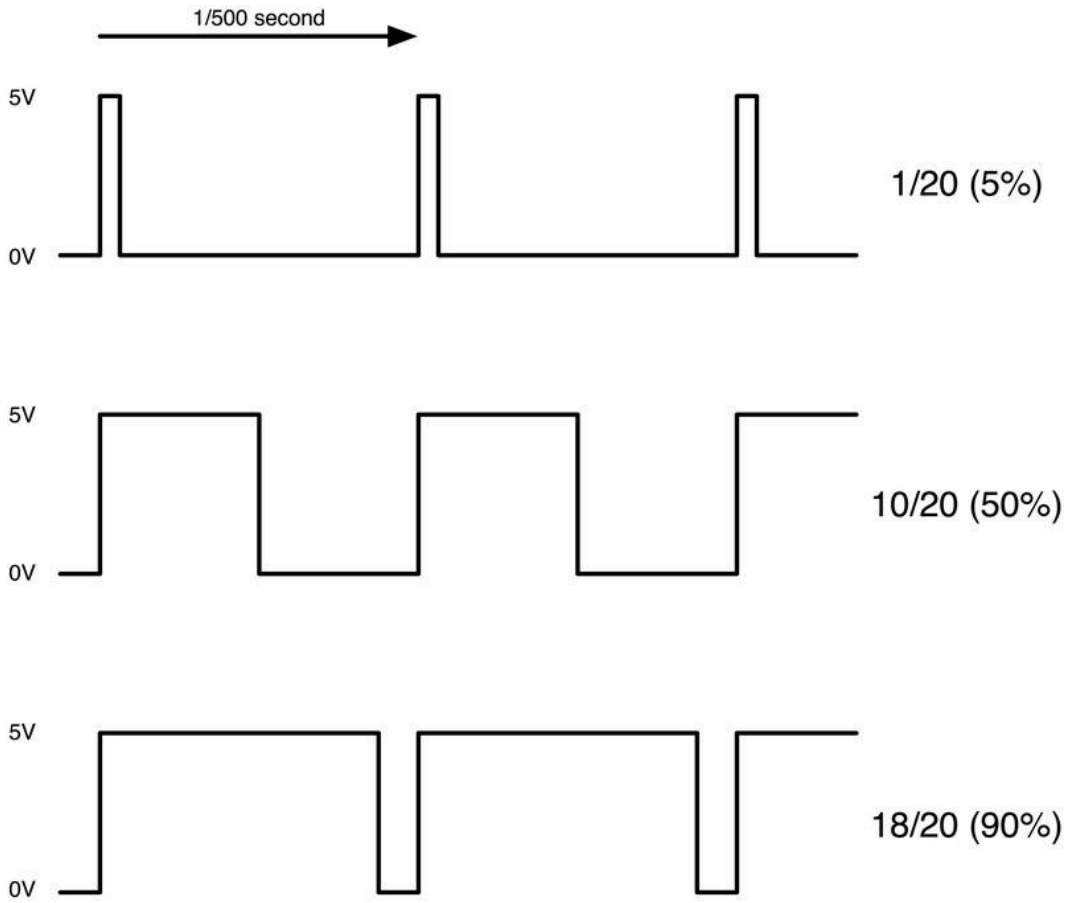


Théorie (PWM)

Pulse Width Modulation (PWM) ou modulation de largeur de pulsation est une technique pour contrôler la puissance d'une sortie.

Vous pouvez l'utiliser pour contrôler la luminosité de chaque led.

Les diagrammes suivants illustrent le fonctionnement des sorties PWM de la carte UNO R3:



Toutes les 1/500 de seconde, les sorties PWM vont produire une impulsion. La longueur de ces impulsions peut être contrôlée avec la fonction 'analogWrite'. Ainsi 'analogWrite(0)' ne produira pas d'impulsion et 'analogWrite(255)' produira une impulsion qui durera jusqu'au prochain déclenchement. Il est donc possible en ajustant la valeur entre 0 et 255 de moduler la puissance en sortie en 0% et 100%

Si la sortie est de 5V sur 100% du temps, en modulant à 90%, nous obtiendrons 90% de 5V.

Connexion

Schéma de câblage

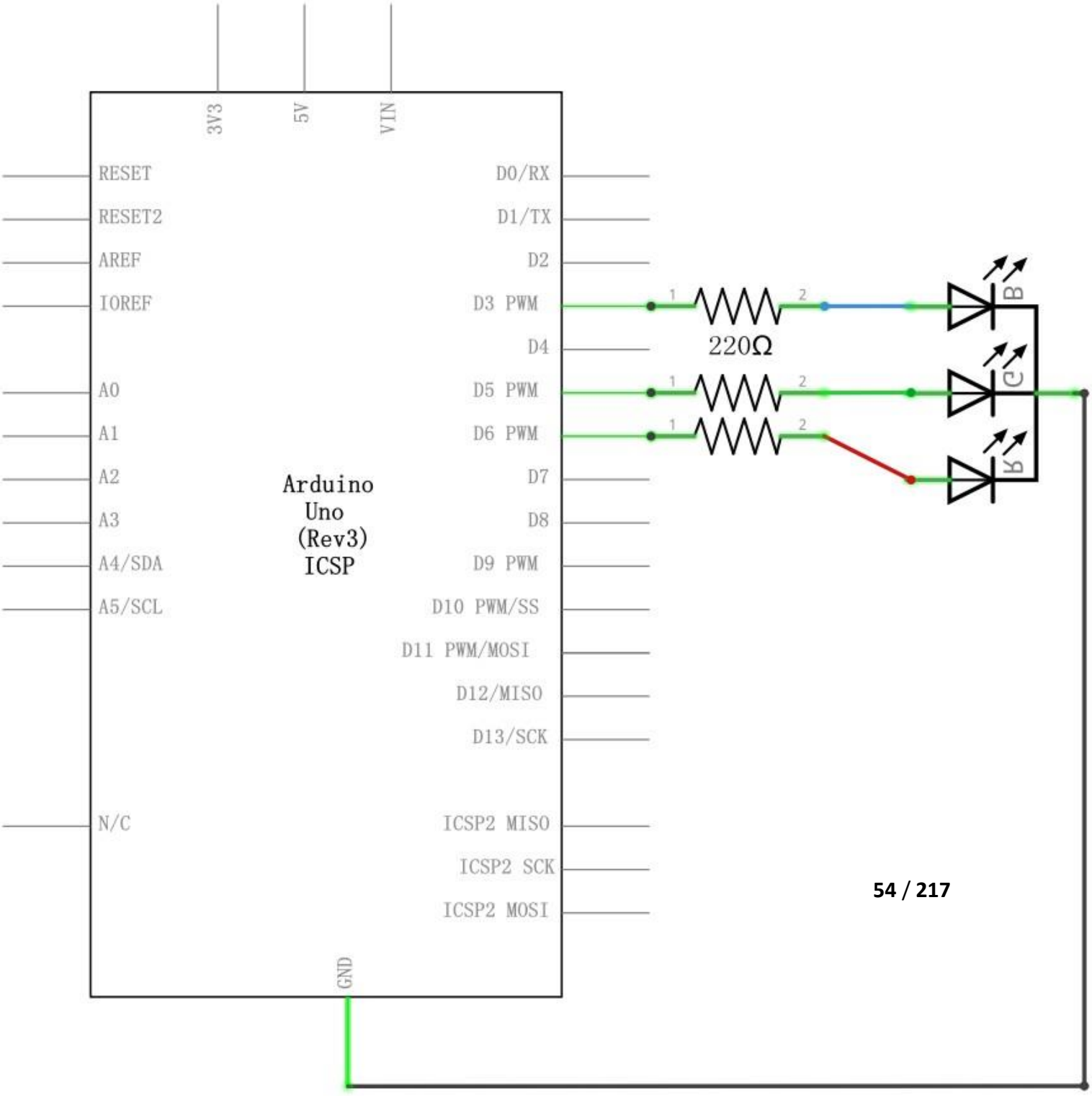
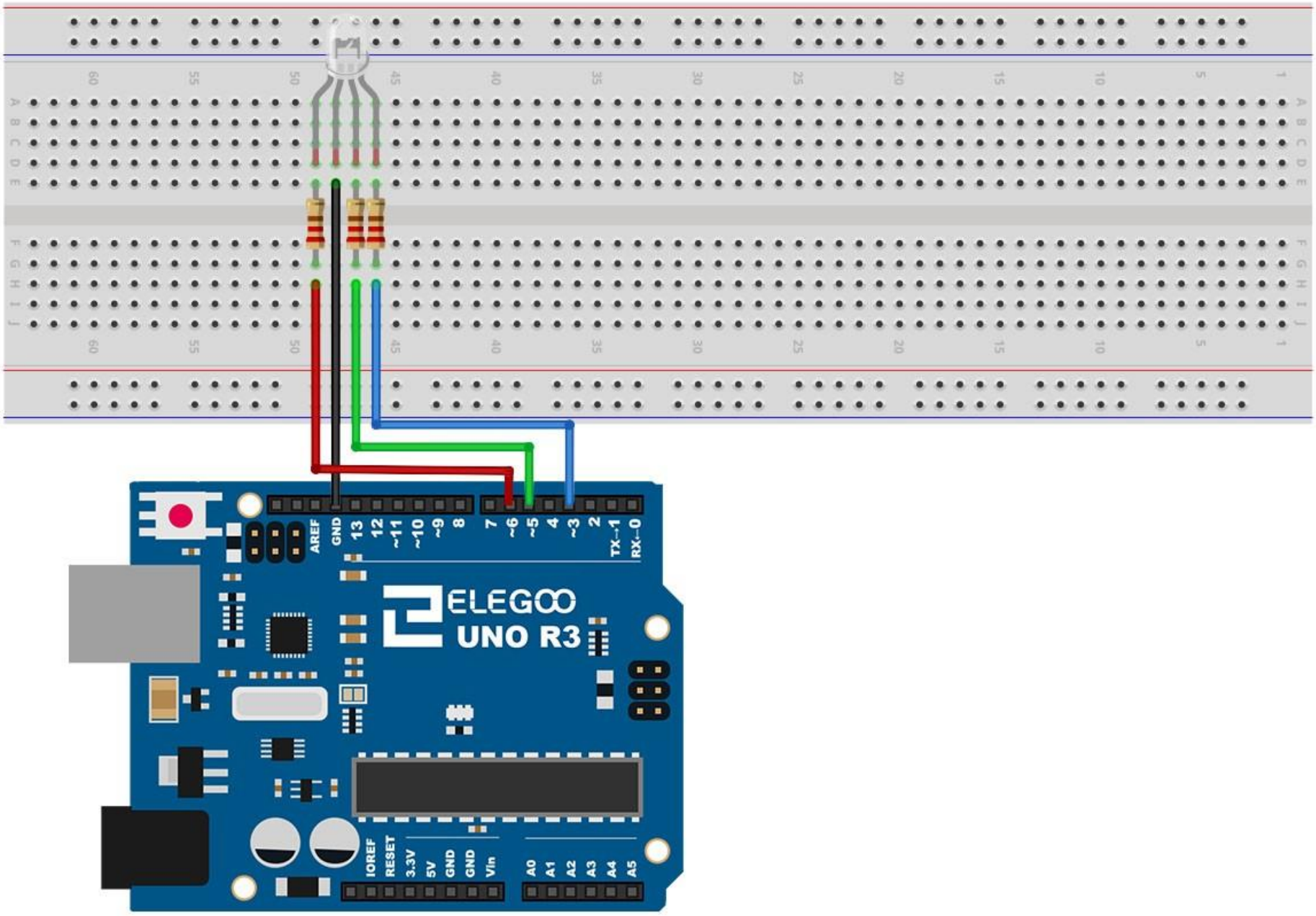
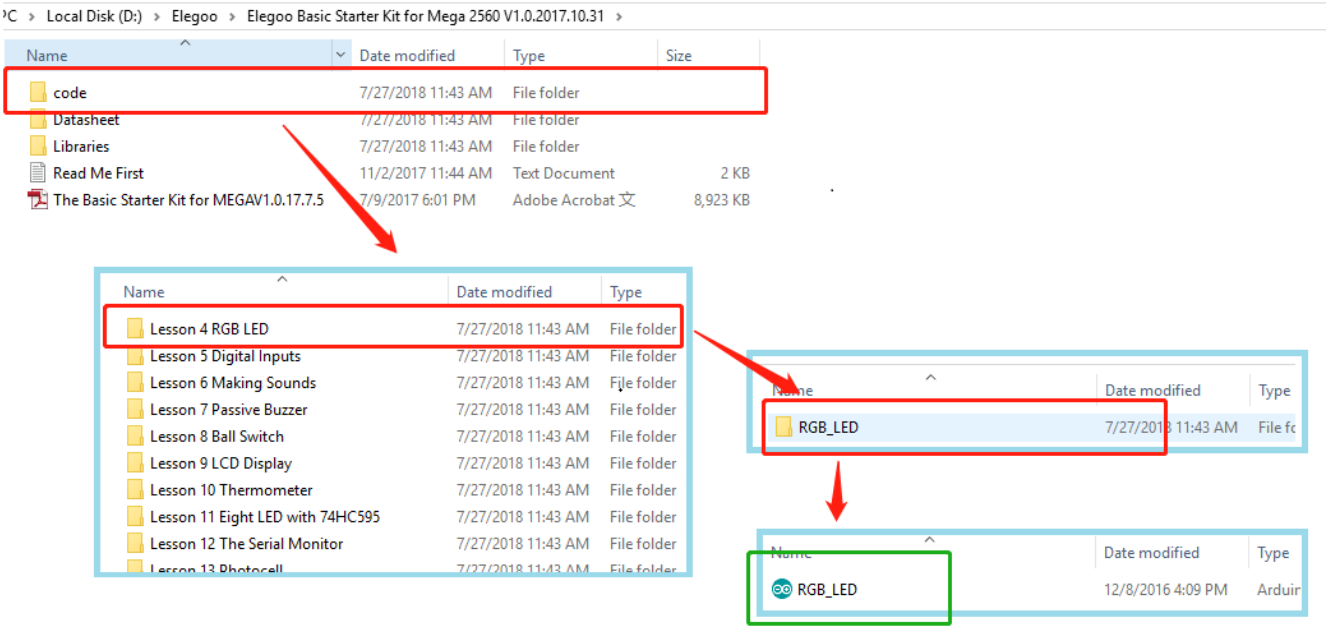


Diagramme de câblage



Code

After wiring, please open the Sketch in folder path: Tutorial > English > code > Lesson 4 RGB LED > RGB_LED, and click UPLOAD to upload the program.



Le sketch commence en spécifiant les pins utilisées pour chaque couleur:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

Le bloc “setup” associe les pins à leur couleur et initialise les pins en tant que sorties analogiques :

```
void setup()
{
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);
digitalWrite(RED, HIGH);
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);
}
```

Avant de regarder le bloc “loop” en détail, regardons la dernière fonction du sketch

```
redValue = 255; // choose a value between 1 and 255 to change the color.
greenValue = 0;
blueValue = 0;
```

Cette fonction prend en compte 3 arguments. Un pour la brillance du rouge, un pour celle du vert et enfin un pour celle du bleu. Dans chaque cas, la valeur à assigner

varie entre 0 et 255. La fonction appelle ensuite analogWrite pour définir la brillance de chaque led.

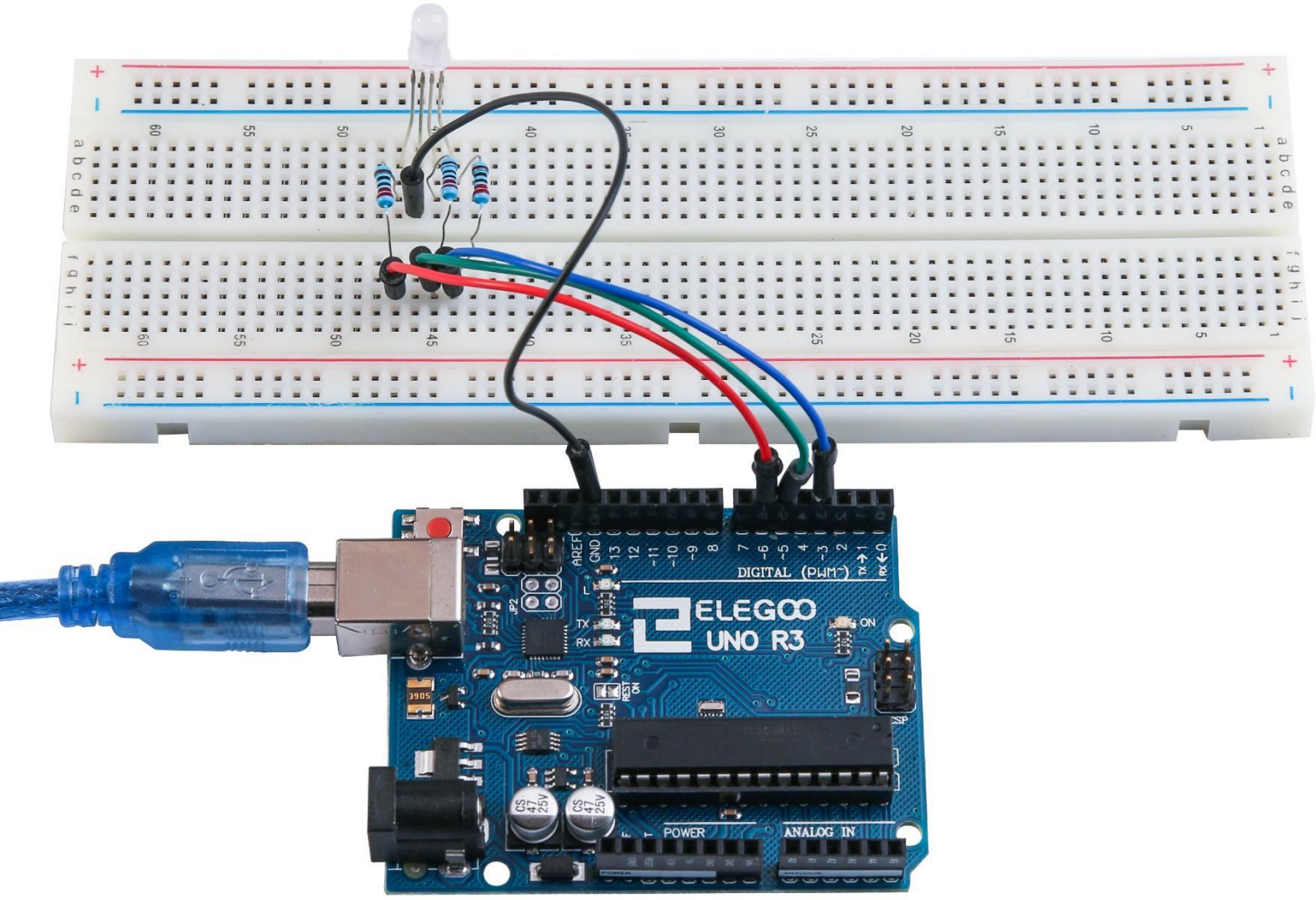
Si vous regarder le bloc “loop” vous constatez que l’on définit la quantité de rouge, de vert, de bleu et que l’on fait une pause avant de passer à la suivante.

#define delayTime 10 // fading time between colors

Delay(delayTime);

Essayez d'ajouter quelques couleurs de votre choix à l'esquisse et regardez l'effet sur votre LED.

Illustration



Leçon 5 Digital Inputs

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser les entrées digitales pour allumer/éteindre une led.

Presser un premier bouton allumera la led, presser un autre bouton l'éteindra.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x LED rouge 5mm

(1) x résistance 220 ohms

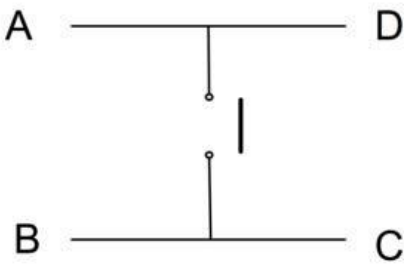
(2) x boutons poussoirs

(7) x Câbles mâle-mâle

Présentation du composant

Boutons poussoirs:

Les boutons poussoirs sont des composants très simples. Lorsque vous pressez le bouton, un contact électrique se fait et laisse passer le courant. Les boutons poussoirs utilisés ont 4 pattes, ce qui peut créer une certaine confusion.



En fait, il n'y a bien que 2 connexions électrique. A et D sont connectées ensemble et B et C aussi. Presser le bouton permet au courant de lier électriquement A-D avec B-C.

Connexion

Schéma de câblage

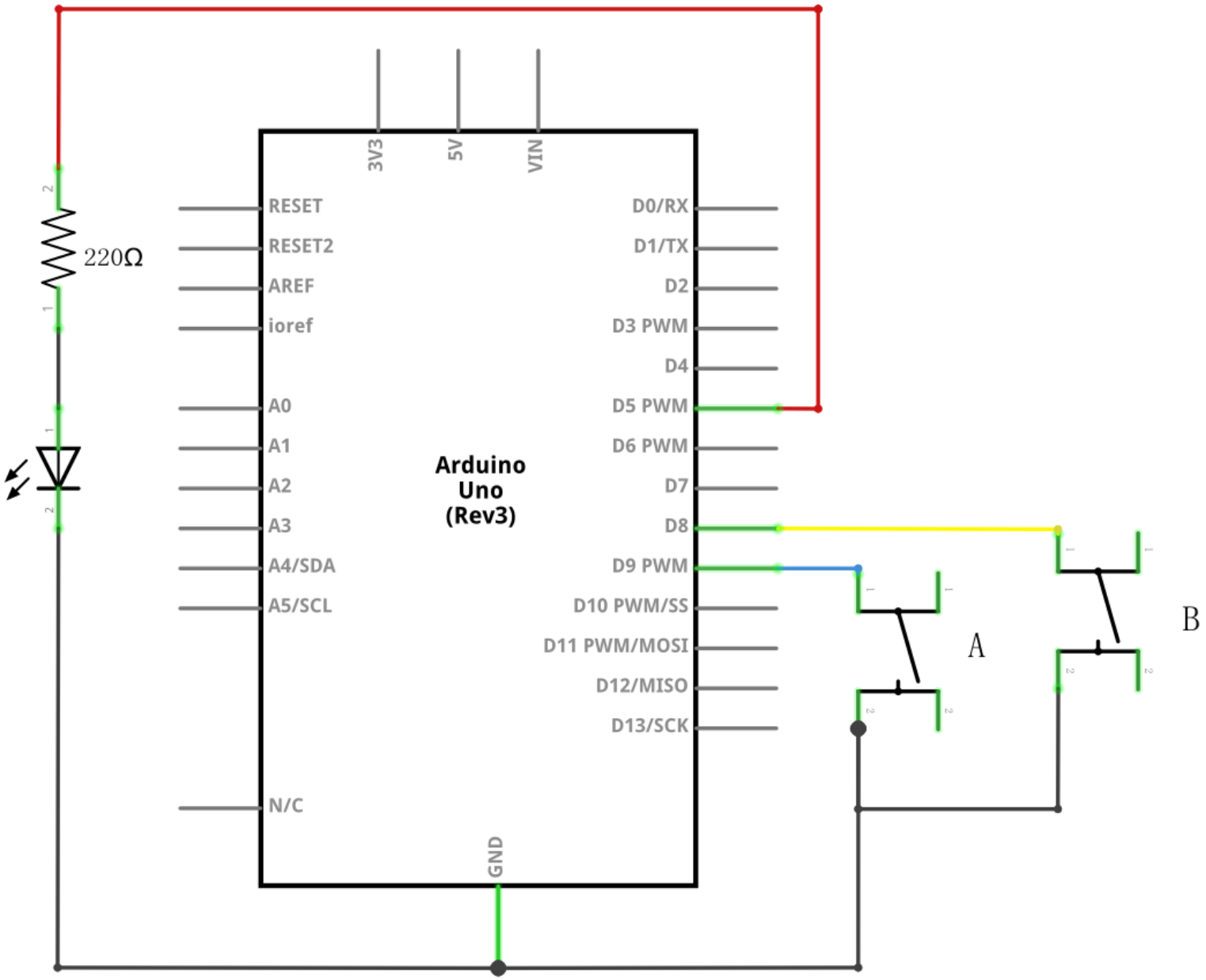
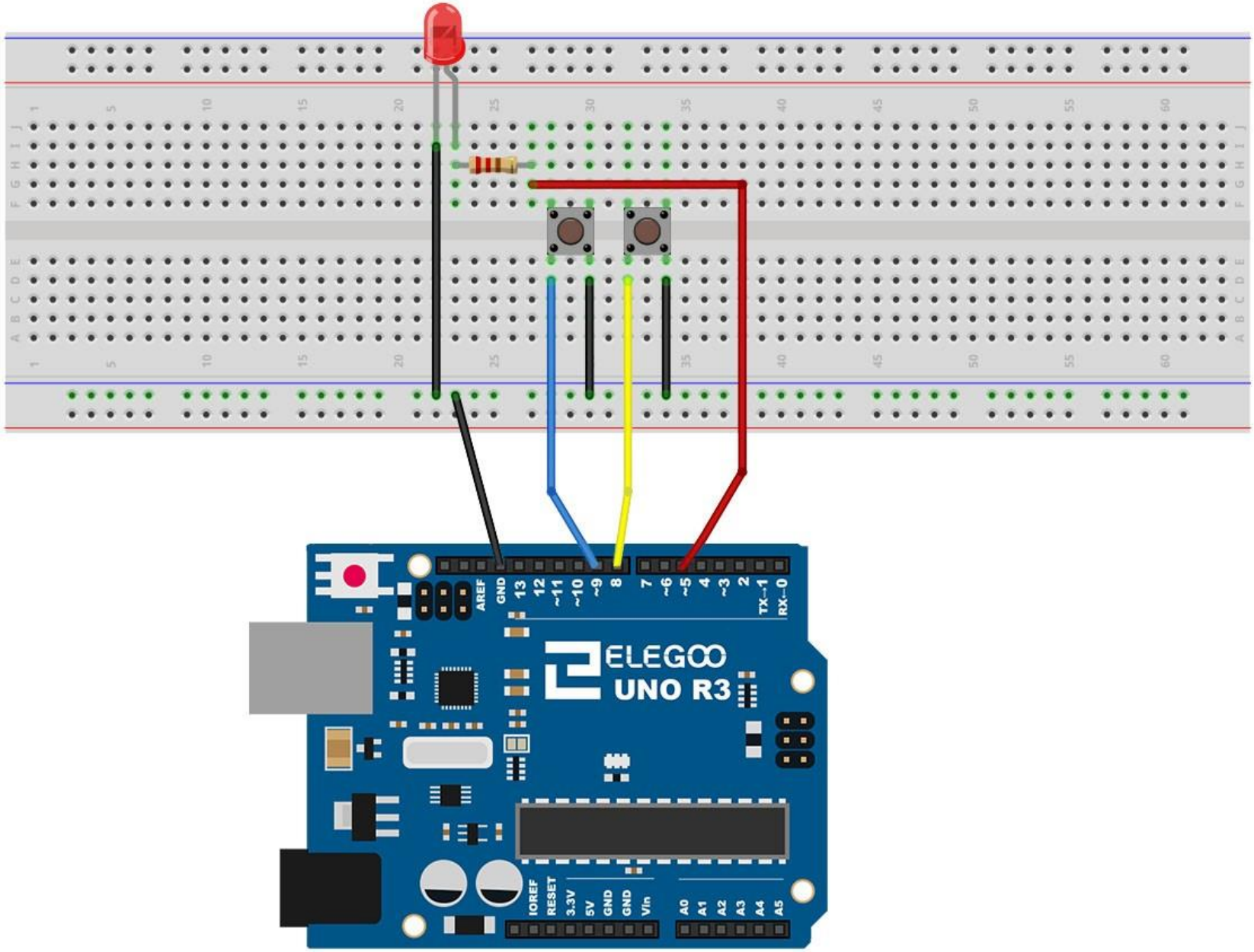


Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 5 Digital Inputs” puis téléversez le code sur la carte UNO R3 comme présenté lors de la leçon 2.

Presser le bouton de gauche allumera la led, presser le bouton de droite l’éteindra.

La première partie du sketch est consacrée à la définition des 3 variables pour les 3 pins qui seront nécessaires au fonctionnement du montage.
Dans le bloc « setup », les pins sont affectées en tant qu’entrées ou sorties.
‘ledPin' est défini en tant que sortie (OUTPUT), 'buttonApin' et 'buttonBpin' en tant qu’entrées (INPUT).

```
pinMode(buttonApin, INPUT_PULLUP);
pinMode(buttonBpin, INPUT_PULLUP);
```

Petite particularité, les entrées sont définies en tant qu’entrées avec résistance de PULLUP. Pourquoi cette subtilité ? Comme vous pouvez le constater sur le schéma de câblage, le fait de presser un bouton met la pin associée à la masse. Mais lorsque le bouton n’est pas pressé, il peut subsister des signaux parasites qui peuvent être interprétés par la carte comme une mise à la masse alors que le bouton n’est pas pressé. Pour éviter ces parasites, on utilise la technique dite de « la résistance de « pullup » qui permet de forcer un état haut via une résistance connectée à une borne +. La carte UNO est capable de simuler ce branchement, pour cela, il suffit de déclarer la pin en INPUT_PULLUP au lieu de faire une déclaration INPUT simple.

Le bloc loop prend ensuite la mesure de la position de chaque bouton:

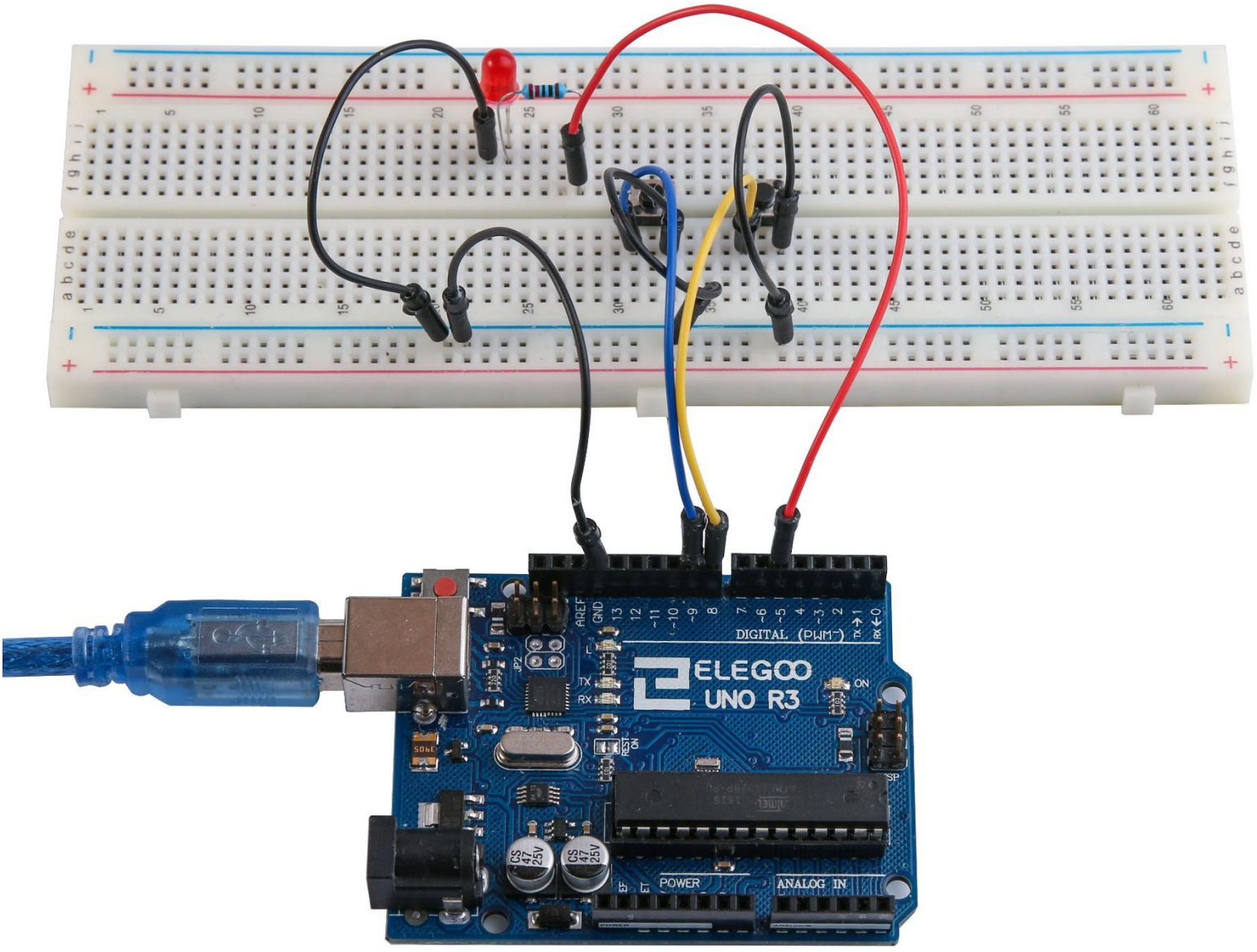
```
void loop()
{
  if (digitalRead(buttonApin) == LOW)
  {
    digitalWrite(ledPin, HIGH);
  }
  if (digitalRead(buttonBpin) == LOW)
  {
    digitalWrite(ledPin, LOW);
```

```
    }
}
```

On regarde si le bouton A est pressé (LOW). Si oui, on allume la led, si non, on ne fait rien.

On regarde ensuite si le bouton B est pressé. Si oui on éteint la led, si non, on ne fait rien.

Illustration



Leçon 6 Active buzzer

But de la leçon

Dans cette leçon, vous allez apprendre à générer des sons avec un « active buzzer ».

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Active buzzer

(2) x Câbles Mâle-Femelle

Présentation du composant

BUZZER:

Les buzzers électroniques sont alimentés en courant continu et équipés de circuits intégrés. Ils sont largement utilisés dans les ordinateurs, imprimantes, alarmes, jouets etc…. Il en existe deux types : les actifs et les passifs. Placez le buzzer avec les pattes vers le haut. Celui où vous pouvez distinguer un petit circuit généralement vert est un buzzer passif.

La différence entre les deux réside dans le fait qu'un buzzer actif possède un oscillateur intégré, il va donc générer un son lorsque le courant passe. Un buzzer passif utilise un signal en entrée pour générer le son (généralement signal carré en 2kHz et 5kHz).

Connection

Schéma de câblage

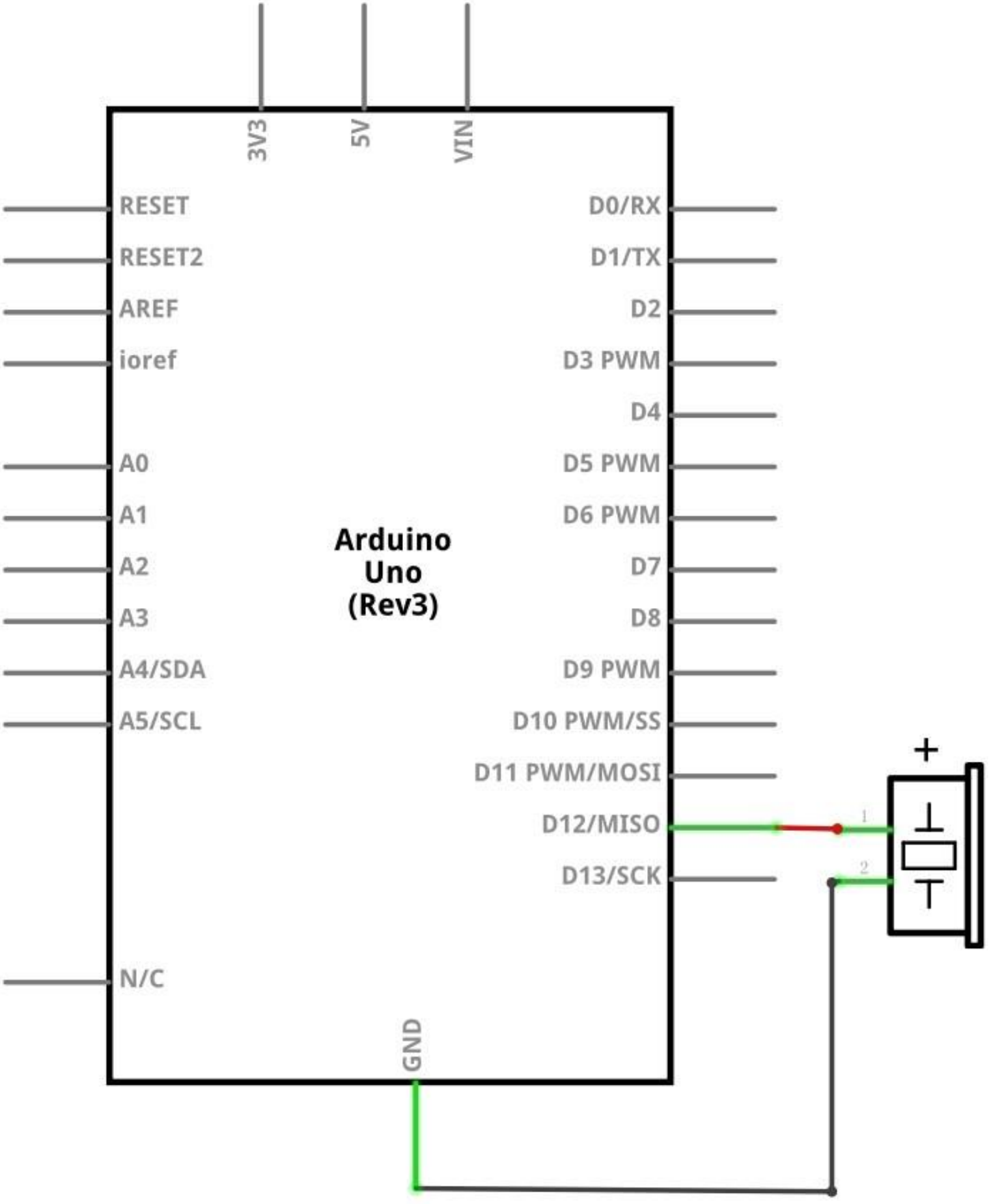
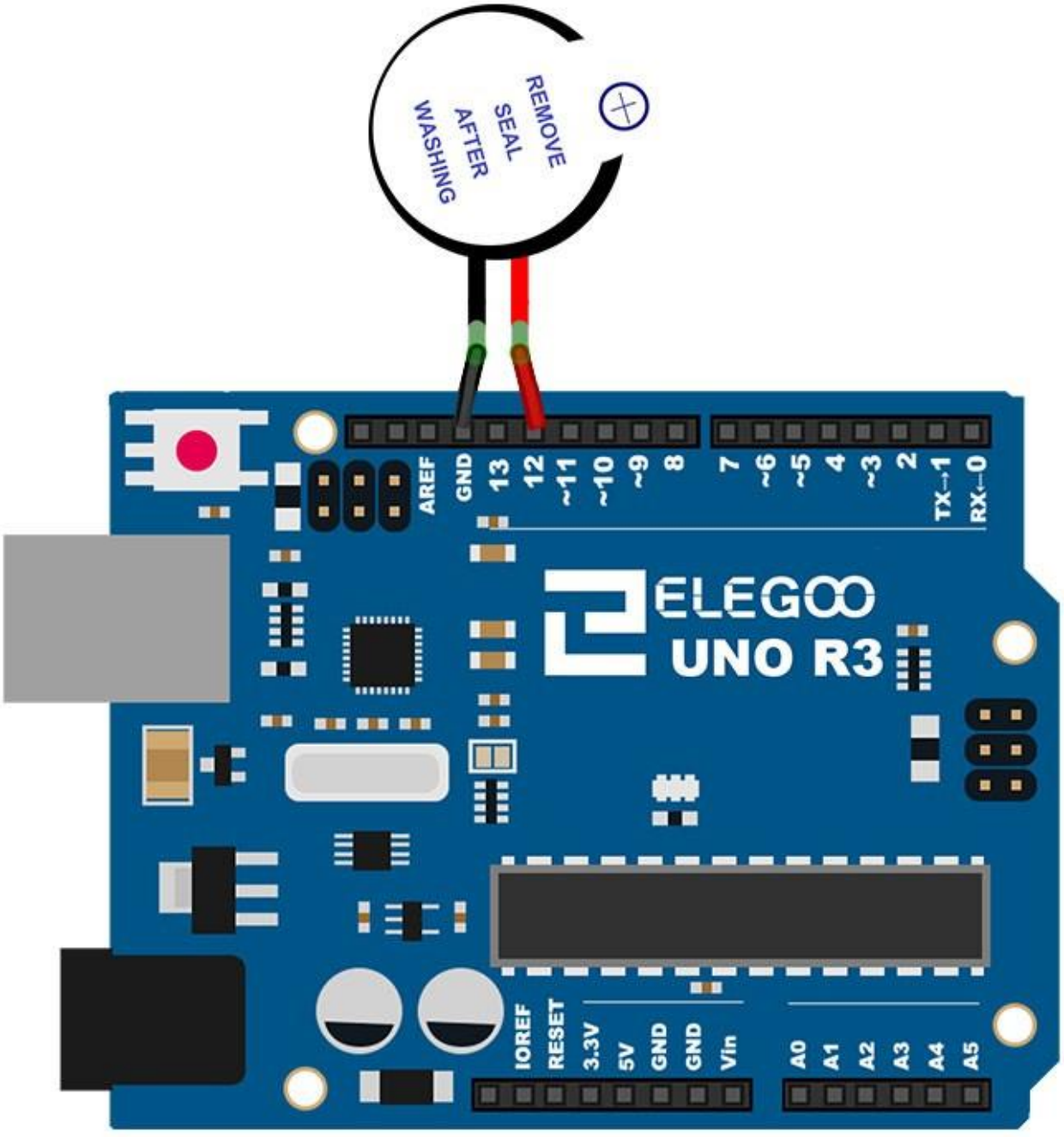


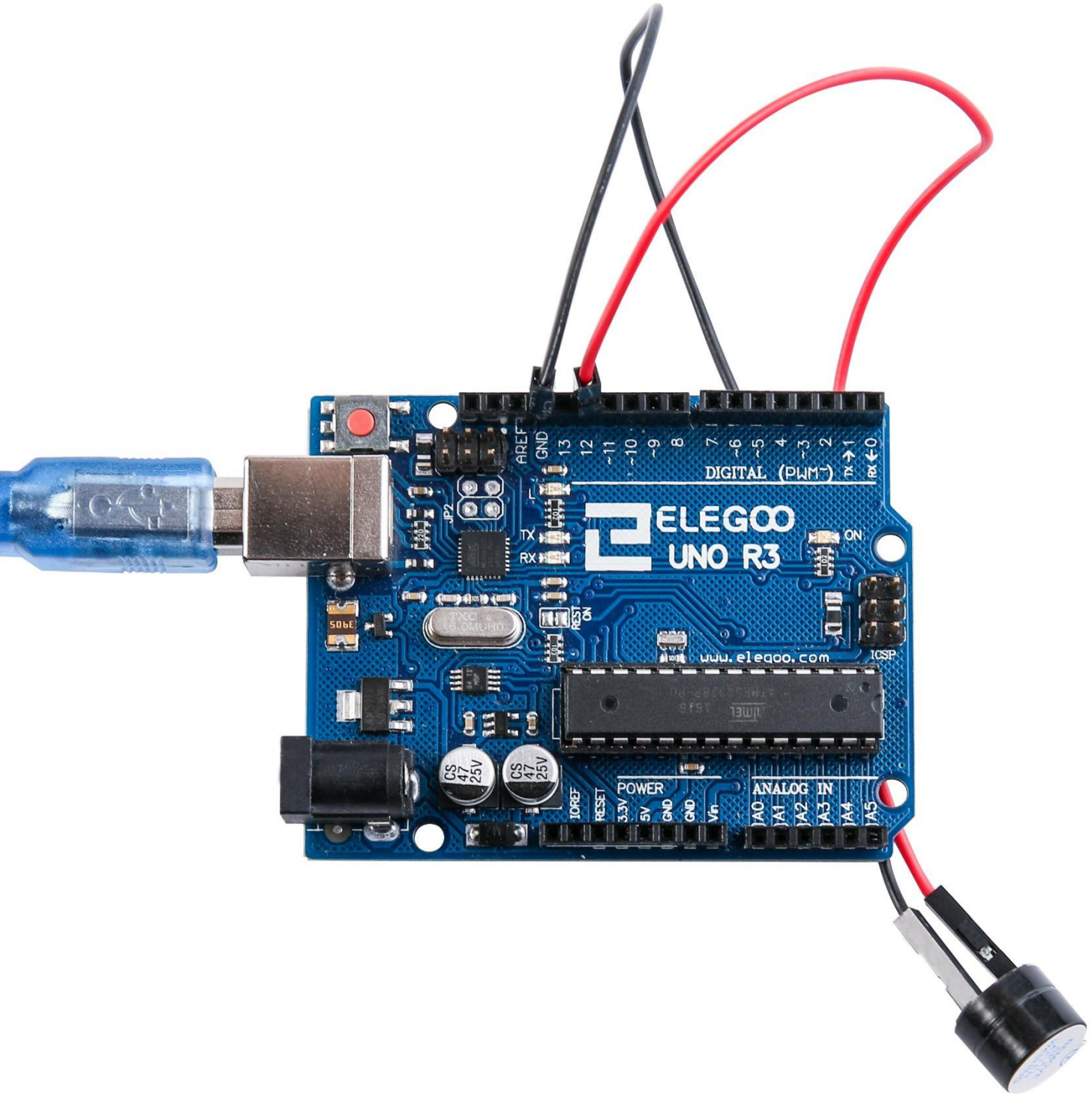
Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 6 Making Sounds” et Téléversez le code sur la carte UNO R3 comme expliqué avec la leçon 2.

Illustration



Leçon 7 Passive Buzzer

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser un buzzer passif.
Le but est de générer des notes durant chacune 0.5 seconde Do (523Hz), Ré (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz).

Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x Buzzer passif
(2) x Câbles Mâle-Femelle

Présentation du composant

Passive Buzzer:

Le principe de fonctionnement d'un buzzer passif est d'utiliser un signal PWM. Modulé à la bonne fréquence, la vibration génère différents sons. Exemple, envoyer une pulsation de 523Hz donne un Do, 587Hzdonne un Ré…
Attention : il n'est pas possible d'utiliser la fonction analogWrite() de la carte UNO car le signal est fixe et vaut 500Hz. Il va falloir faire autrement, c'est ce que nous allons voir dans le code qui suit.

Connection

Schéma de câblage

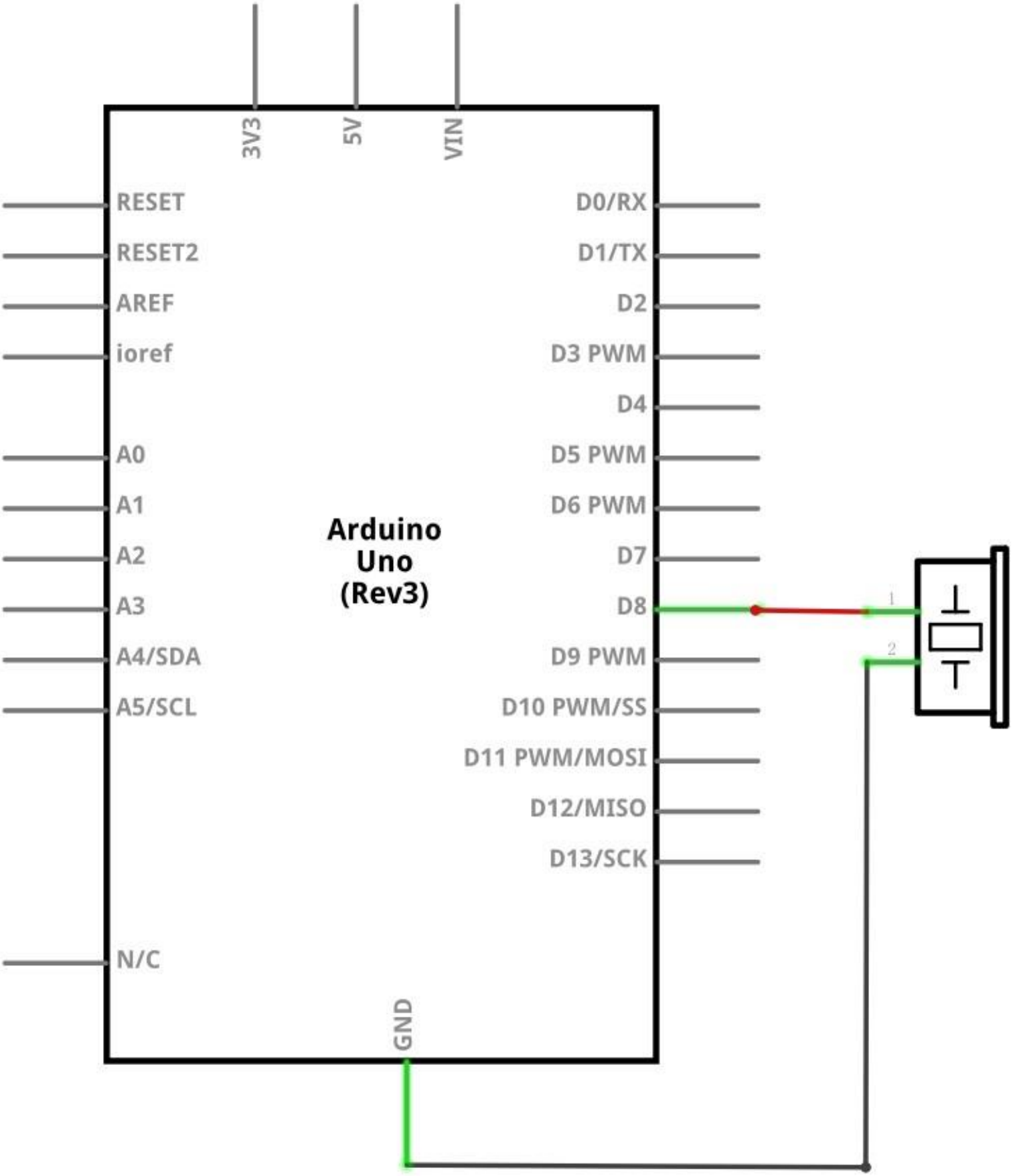
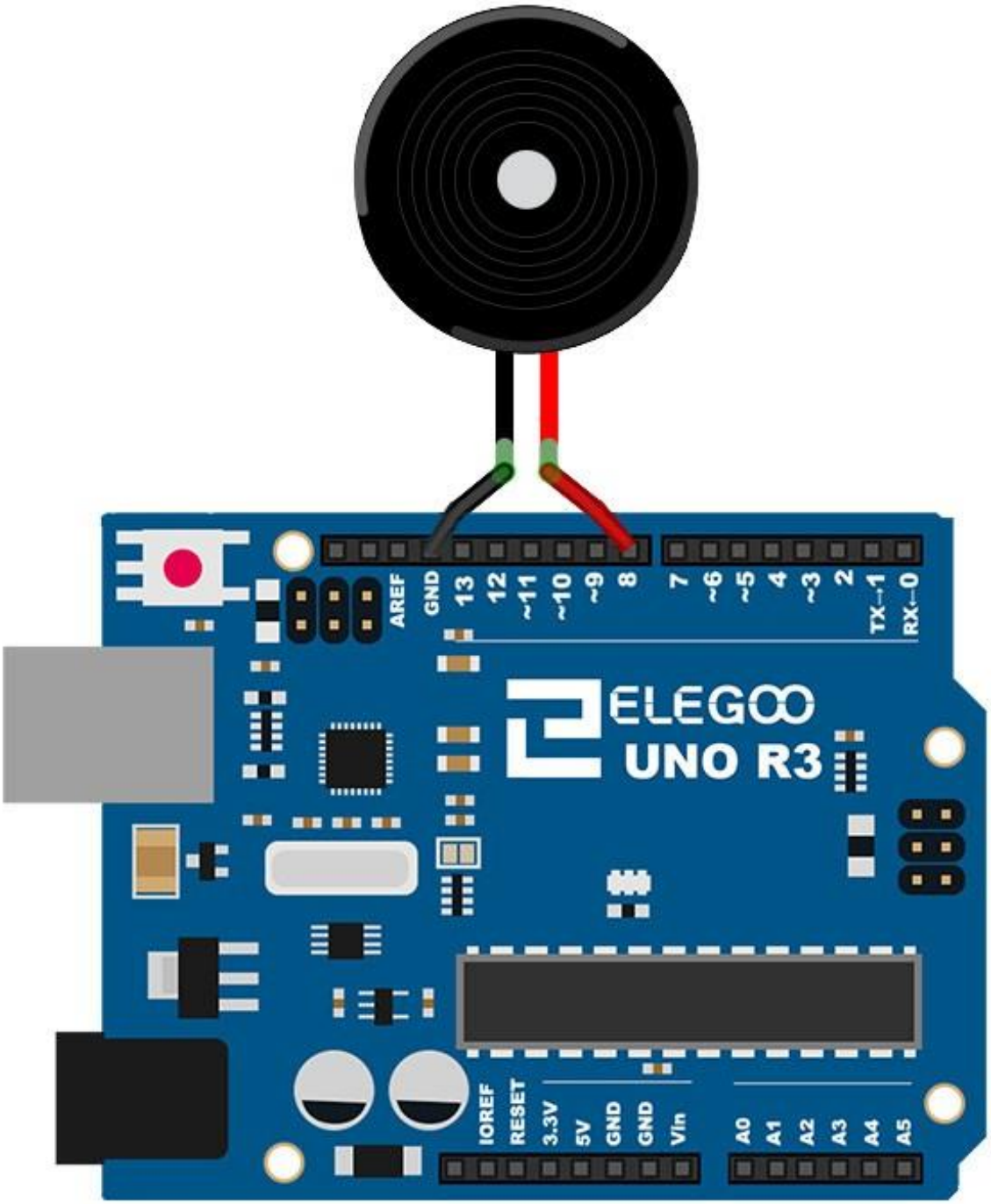


Diagramme de câblage

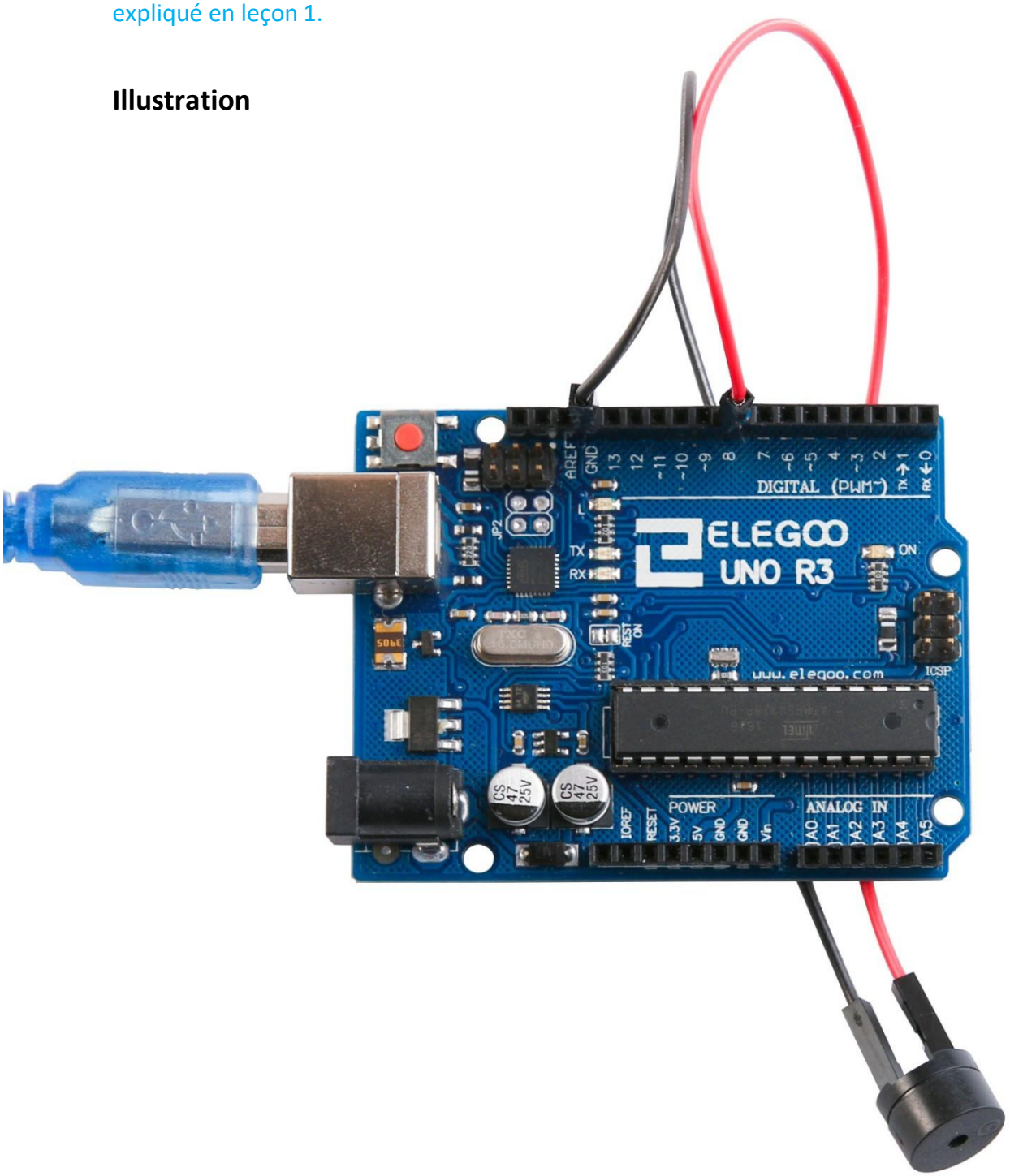


Code

Après avoir réalisé le câblage, ouvrez le sketch ”Leçon 7 Passive Buzzer” et Téléversez-le sur la carte UNO R3.

Attention : veillez d’abord à bien avoir installé la bibliothèque <PITCHES> comme expliqué en leçon 1.

Illustration



Leçon 8 Tilt Ball Switch

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de détecter une inclinaison avec le capteur “tilt ball switch”.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x capteur Tilt Ball switch

(2) x Câbles Mâle-Femelle

Présentation du composant

Capteur « Tilt » :

Ce capteur vous permet de détecter une orientation ou inclinaison. Il est petit, pas cher est consomme très peu d’énergie.

Sa simplicité fait qu’il est très répandu pour les jouets et toutes sortes de gadgets. Vous le trouverez sous différents types et noms : "mercury switches", "tilt switches" ou "rolling ball sensors".

Ces capteurs sont généralement faits d’une cavité cylindrique dans laquelle se trouve une masse libre de mercure par exemple. Au bout de la cavité, deux éléments conducteurs non reliés. Lorsque la cavité change d’orientation ou se retourne, la masse libre vient sur en contact avec les deux éléments conducteur et crée un contact électrique qui laisse passer le courant.

Alors qu’ils ne sont pas aussi précis qu’un accéléromètre, ils sont parfaits pour faire de la détection de mouvement ou d’orientation.

Connection

Schéma de câblage

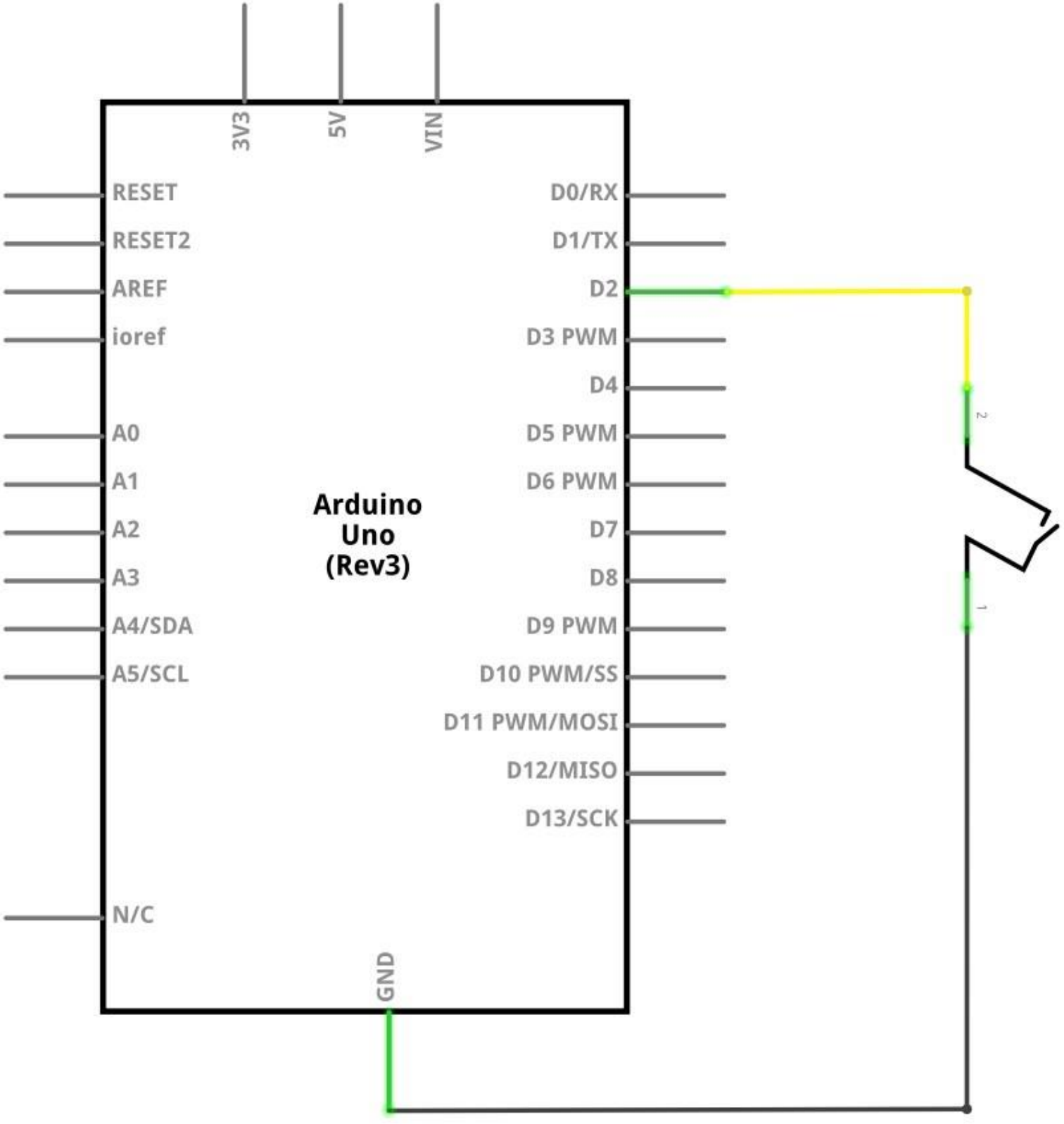
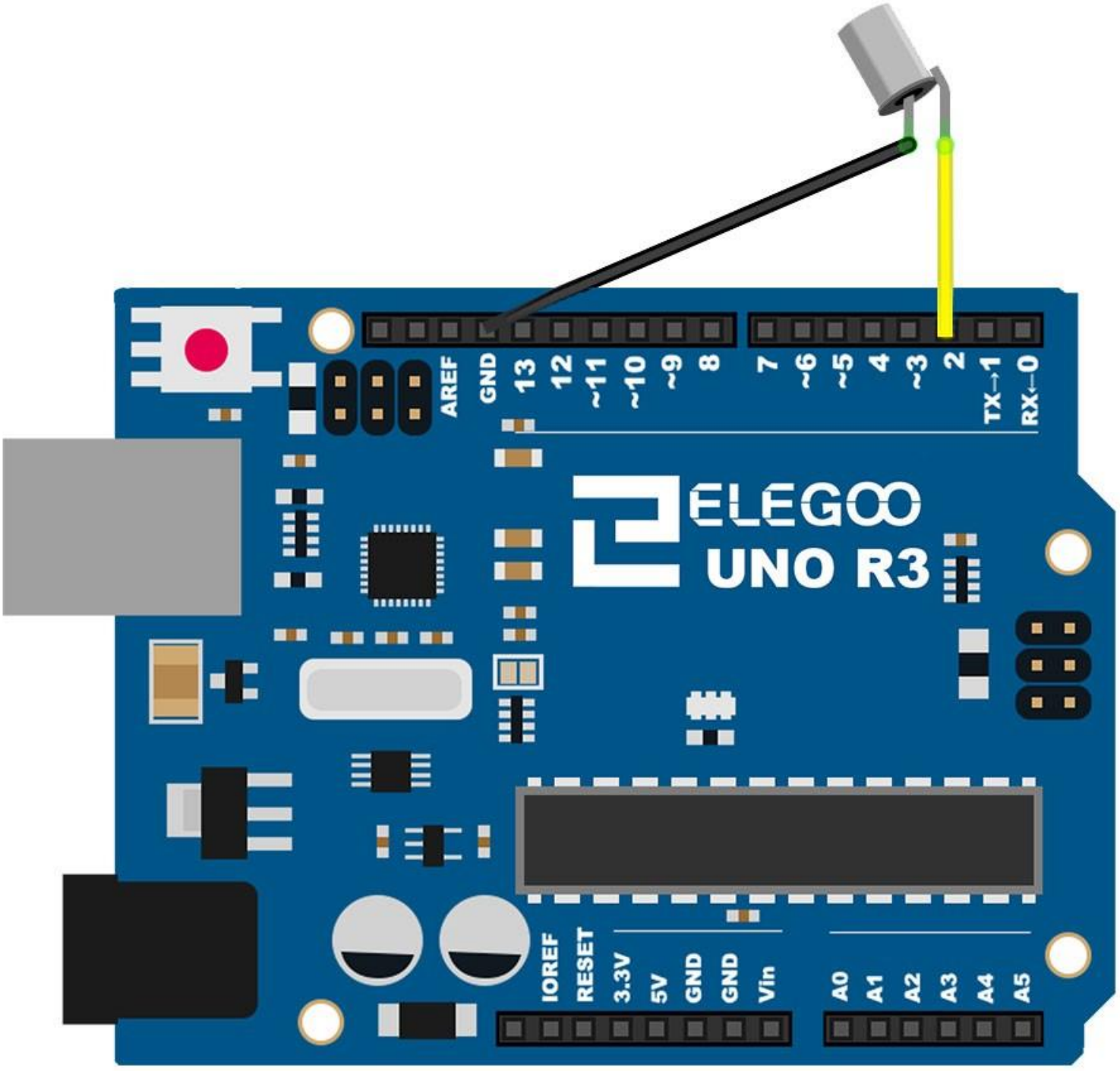


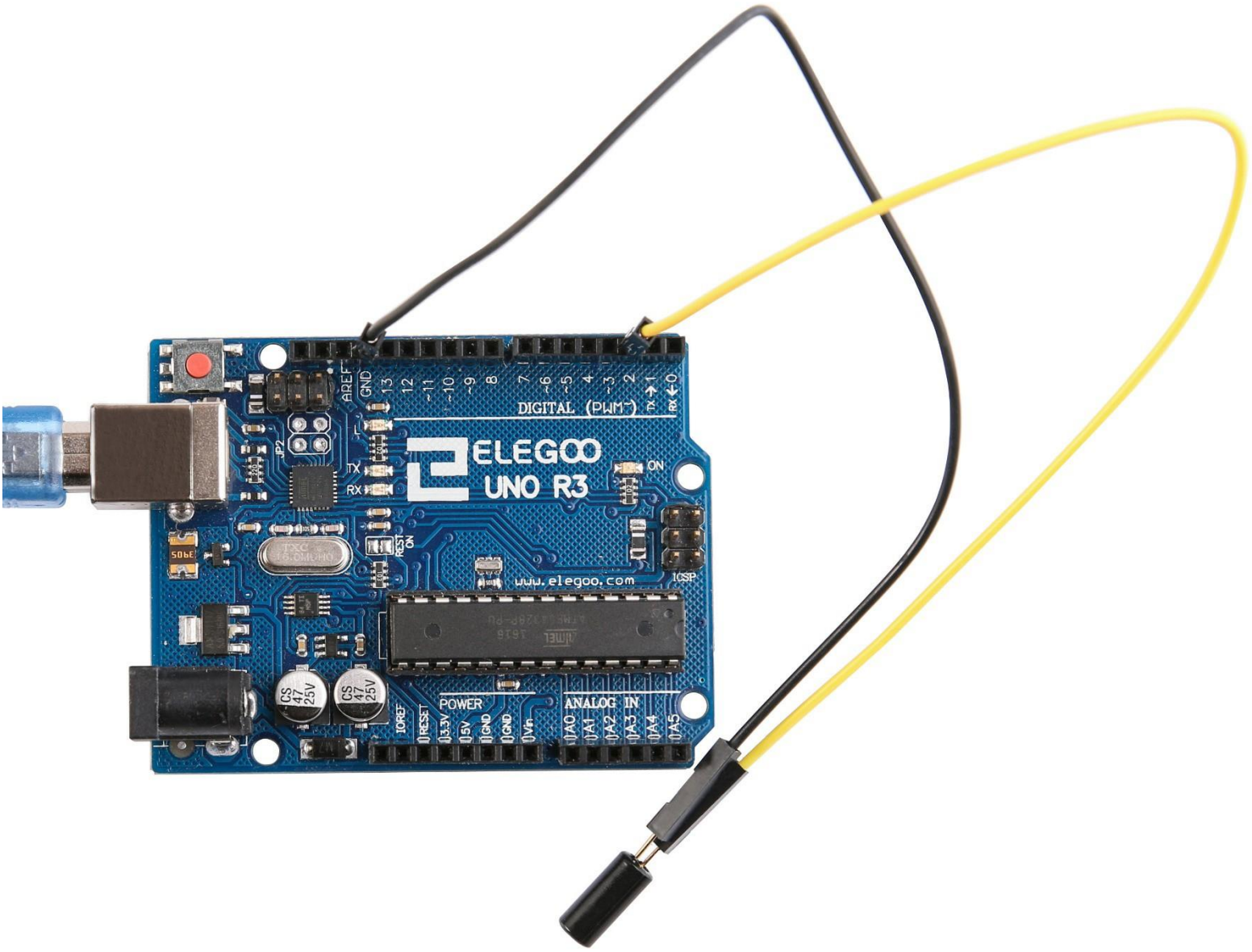
Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 8 Ball Switch” et Téléversez-le sur la carte UNO RO3.

Illustration



Leçon 9 Servomoteur

But de la leçon

Le servomoteur est un type de moteurs qui peut seulement tourner de 180 degrés. Il est contrôlé par l'émission de pulsations électriques depuis la carte UNO R3. La pulsation donne au moteur la position qu'il doit prendre.

Le moteur a trois connexions. Le fil brun (masse), le fil rouge (positif), le fil orange (signal – à connecter sur la pin 9 de la carte UNO R3)

Matériel nécessaire:

(1) x Elegoo Uno R3

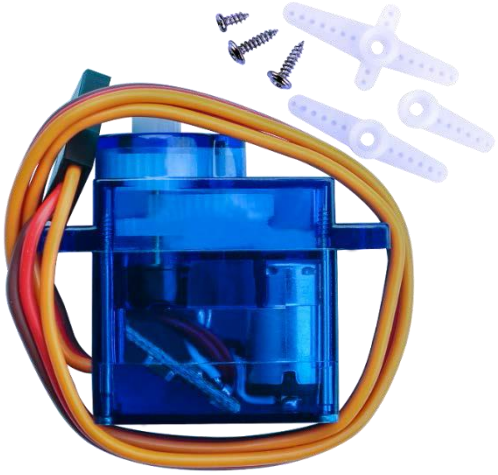
(1) x Servo (SG90)

(3) x Câbles mâle-mâle

Présentation du composant

SG90

- Longueur de câble : 25cm
- Couple (à 4.8V): 1.6kg/cm
- Temperature : -30~60'C
- Voltage: 3.5~6V
- Dimensions : 3.2 cm x 3 cm x 1.2 cm
- Poids : 134 g



Connection

Schéma de câblage

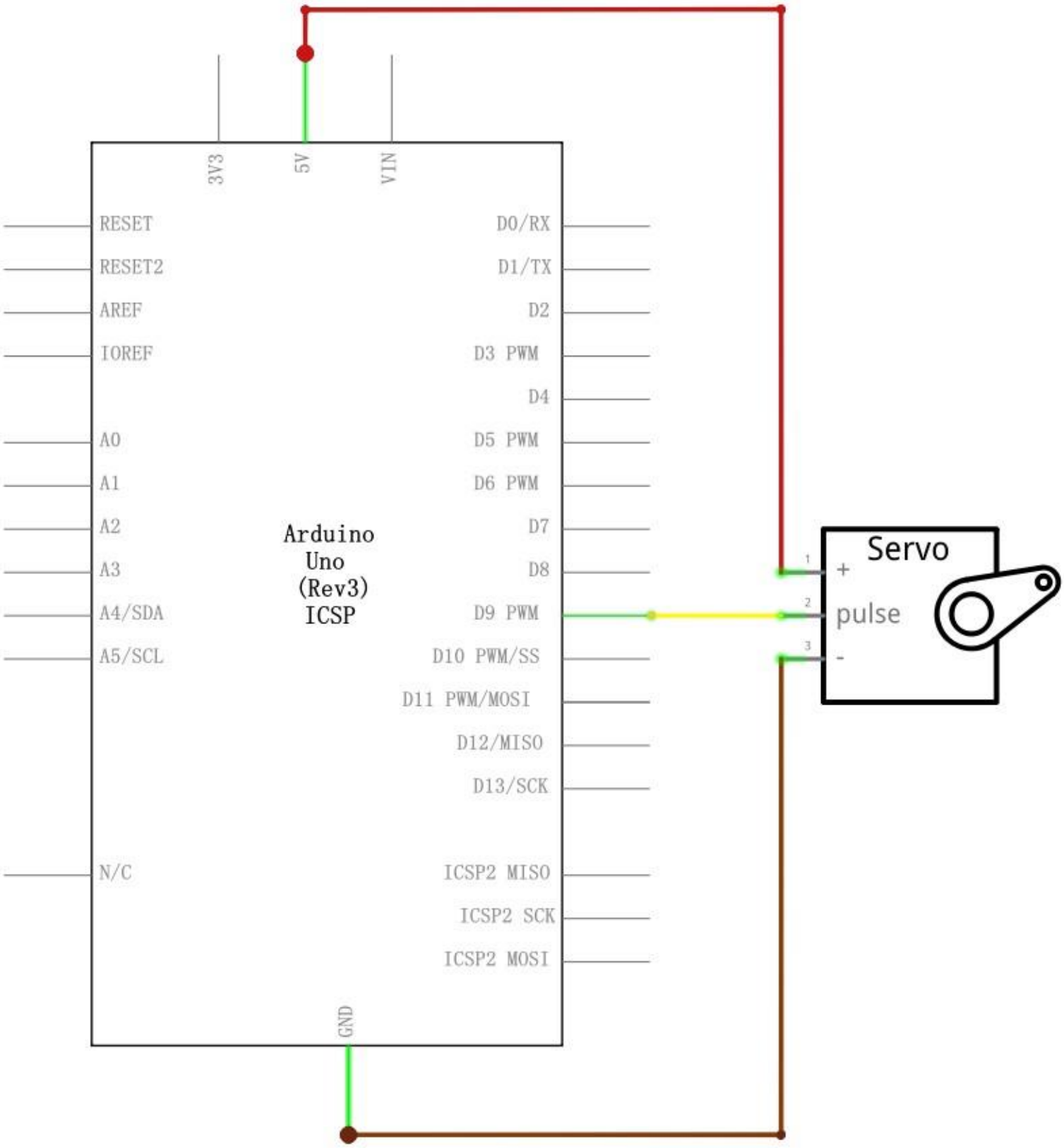
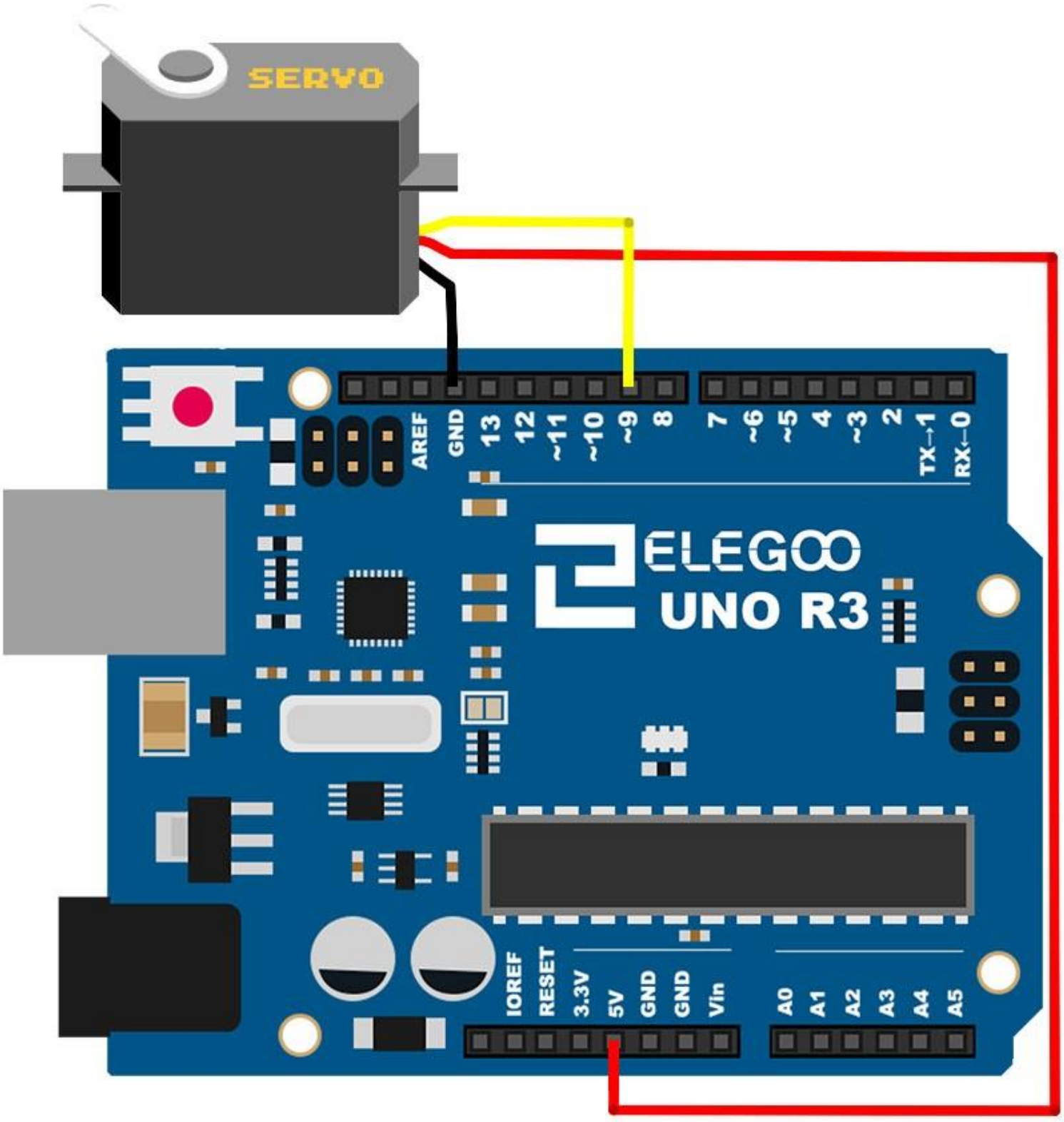


Diagramme de câblage

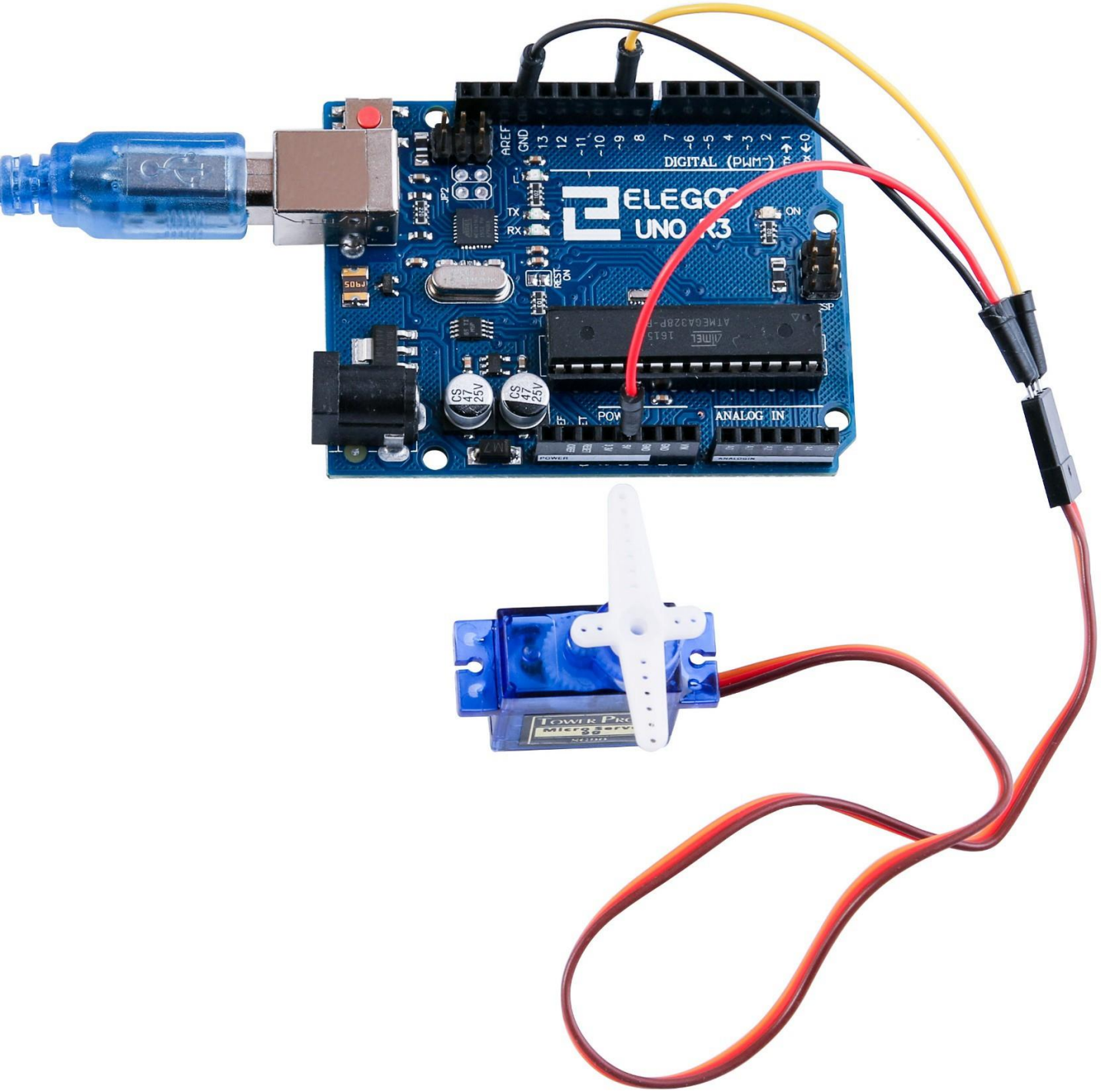


Code

Après avoir réalisé le câblage, ouvrez le sketch « Leçon 9 Servo » et téléversez le code sur la carte comme expliqué à la leçon 2.

Attention de bien avoir installé la bibliothèque <SERVO>, voir la leçon 1 en cas de difficulté.

Illustration

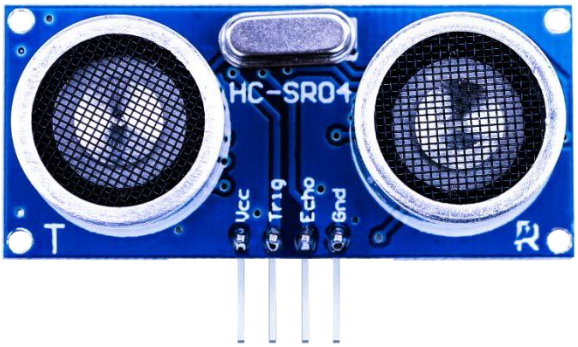


Leçon 10 Ultrasonic Sensor Module

But de la leçon

Le capteur ultrasons est parfait pour tous les projets nécessitant de faire de la mesure de distances, en évitement d'obstacles par exemple.

Le HC-SR04 est un produit peu onéreux et facile à utiliser.



Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Capteur Ultrasons

(4) x Câbles Mâle-Femelle

Présentation du composant

Module Ultrasons

Le capteur permet une mesure d'un objet situé à une distance allant de 2cm à 4m et fourni une mesure avec une précision de 3mm.

Principe de fonctionnement:

(1)Un signal de déclenchement (trigger) est émis pendant 10µs

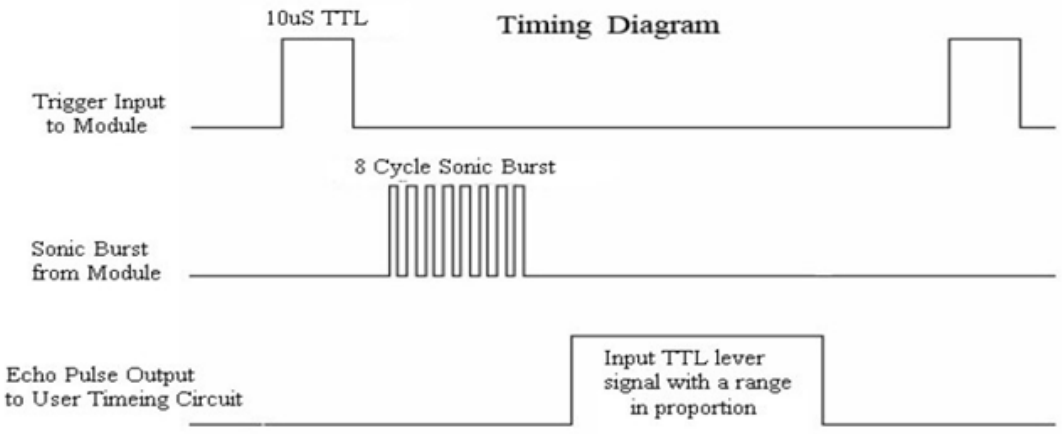
(1)Le module émet 8 signaux (minimum) à 40kHz.

(2)Le module se met à l'écoute d'un signal de retour

(3)Le temps entre l'émission et la réception est le temps nécessaire au signal pour faire l'aller et le retour vers l'objet qui a réfléchi celui-ci.

On a donc :

Distance = (Temps_mesuré * Vitesse_du_son)/2 (340m/s pour la vitesse du son)



Connection

Schéma de câblage

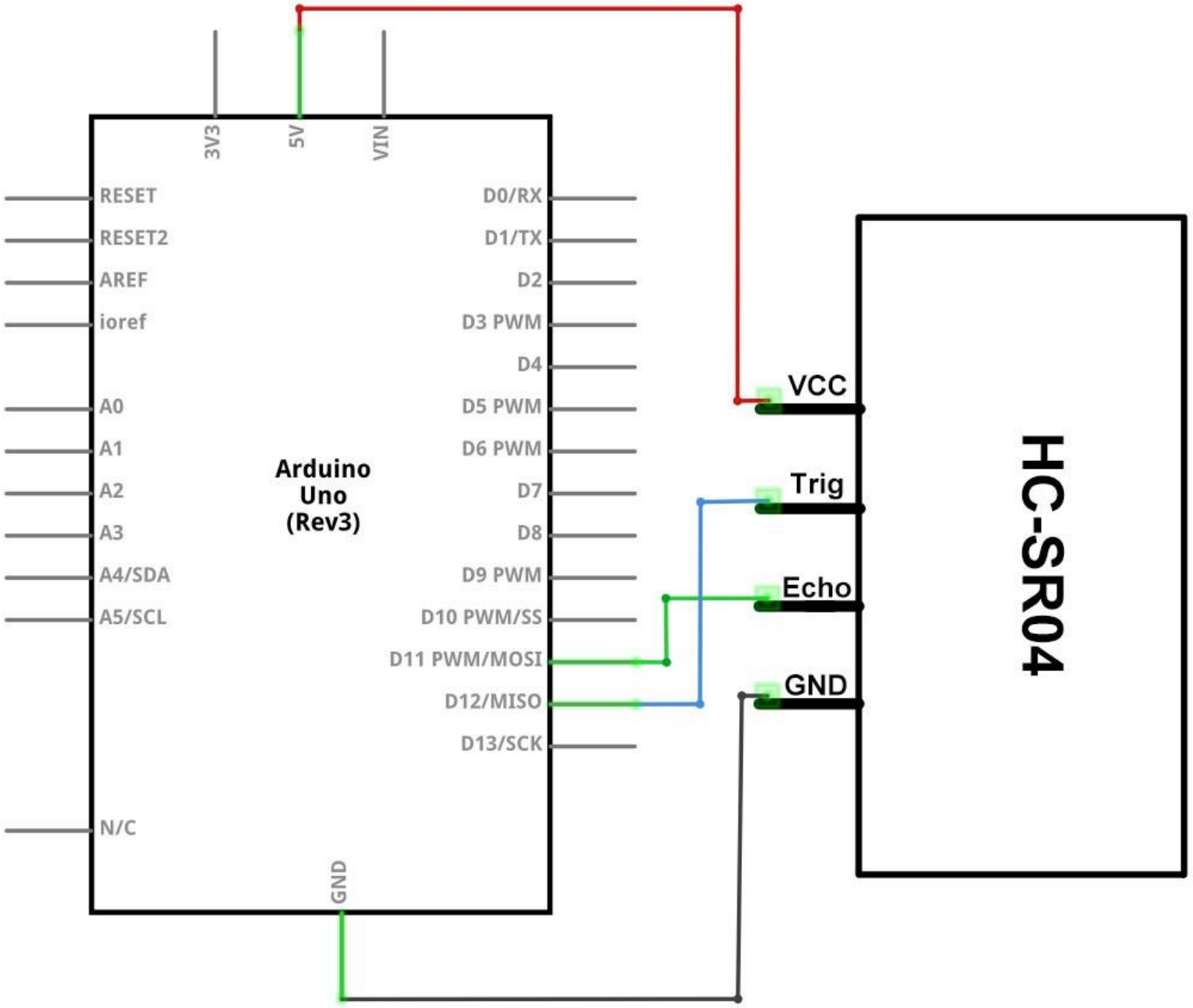
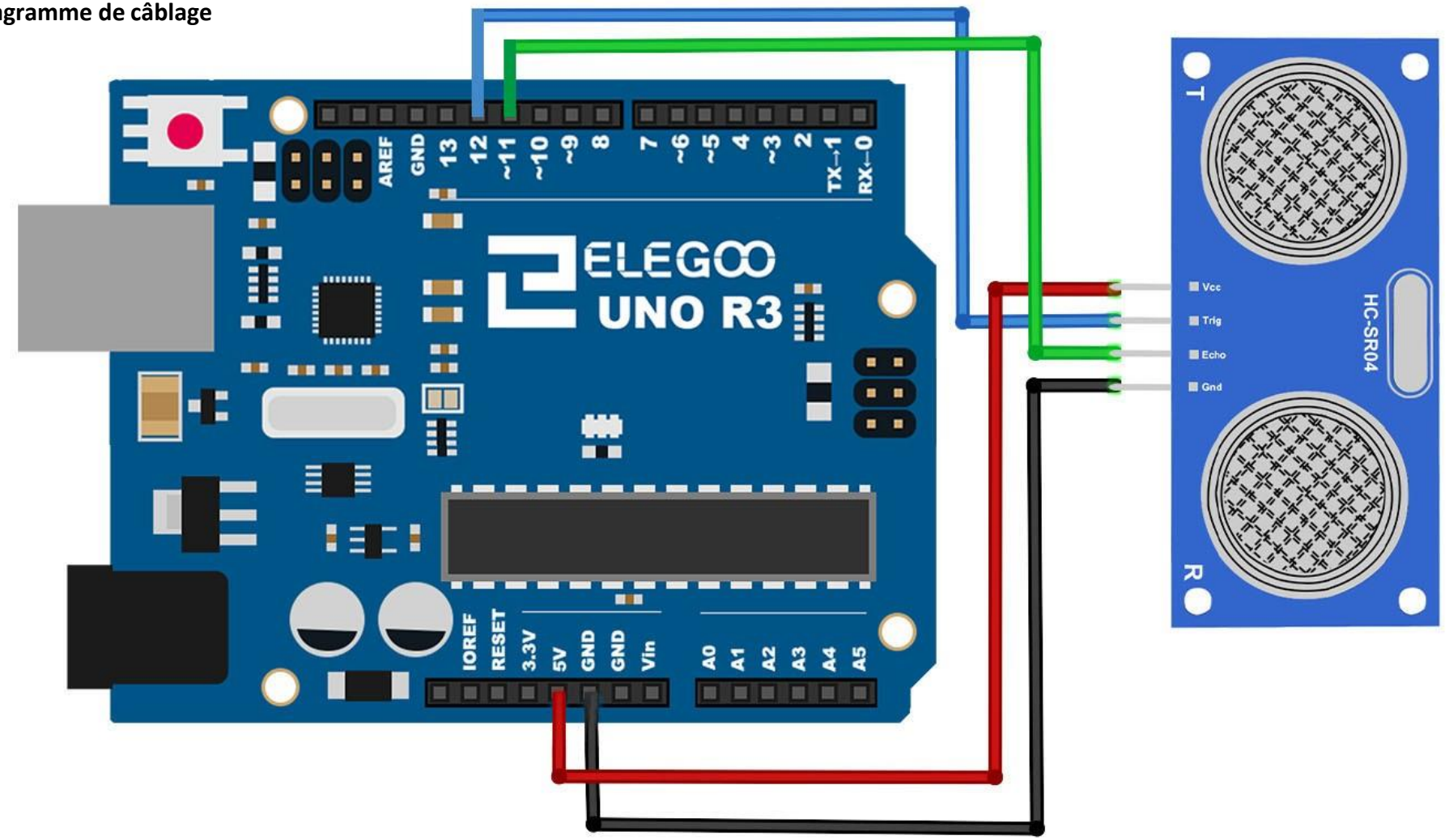


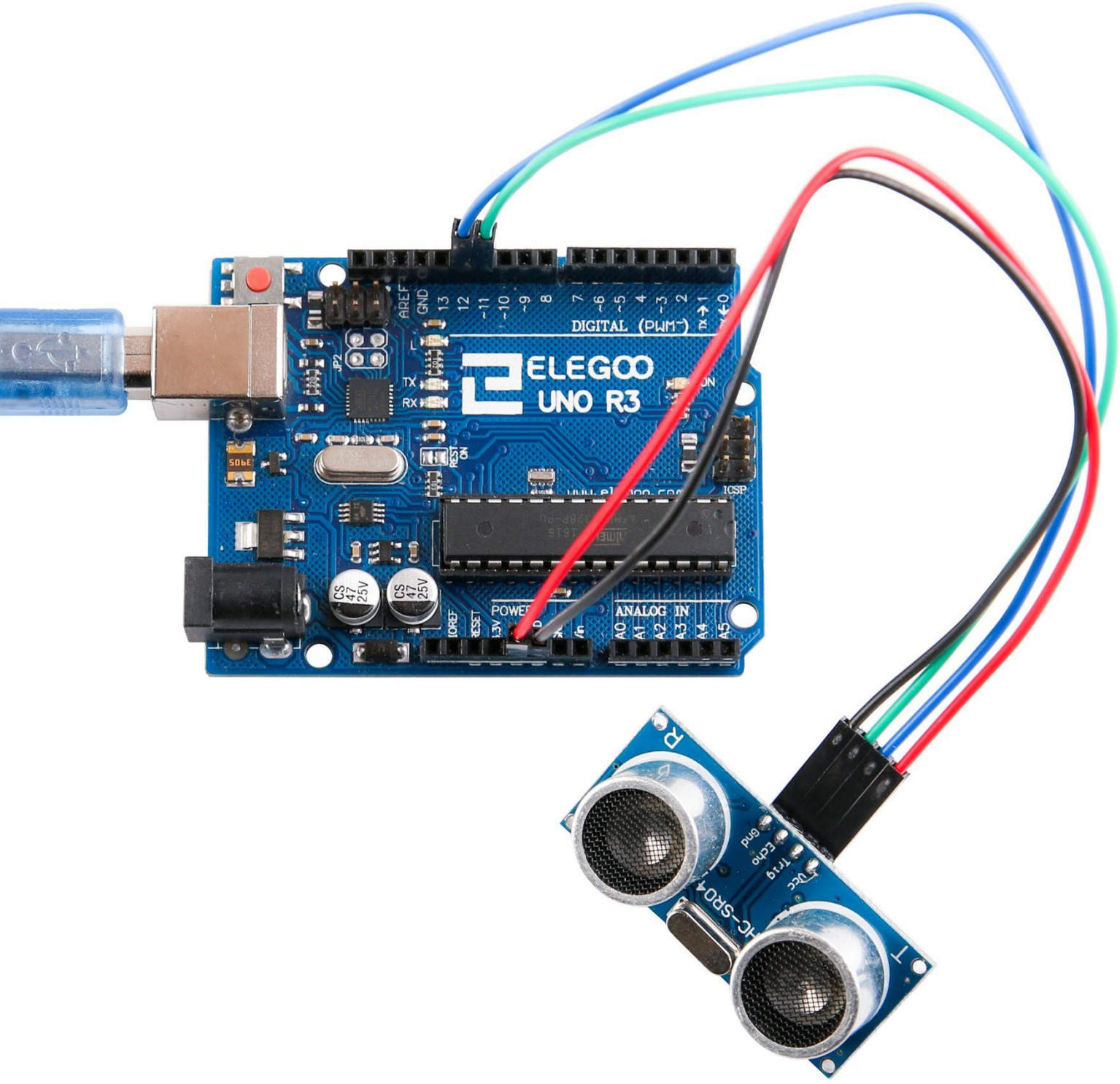
Diagramme de câblage



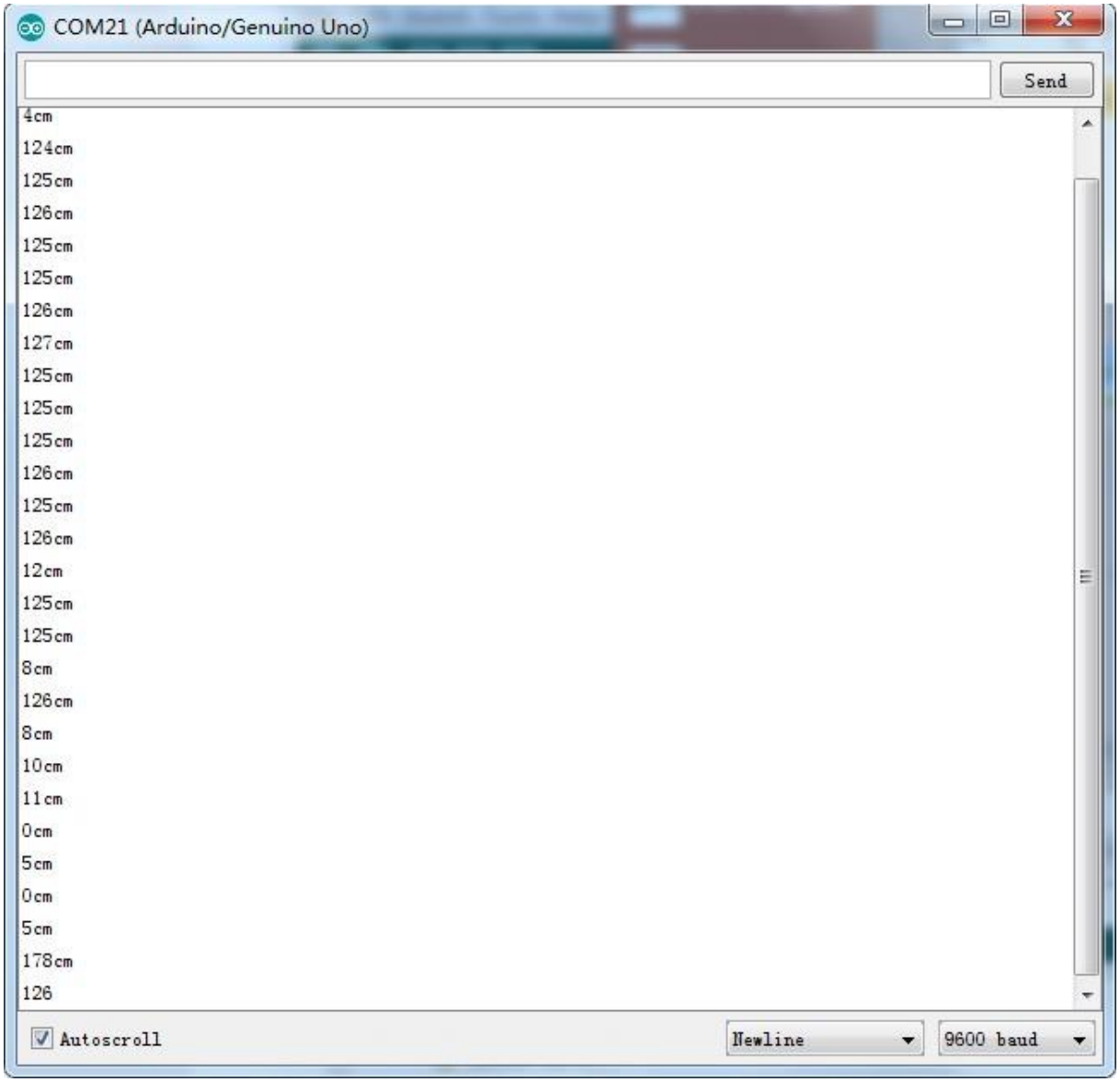
Code

Ouvrez le sketch “Leçon 10 Ultrasonic Sensor Module” et Téléversez-le sur la carte UNO R3.

Illustration



Ouvrez le moniteur série. Les distances mesurées s'affichent au fur et à mesure.



Leçon 11 Membrane Switch Module

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible d'intégrer un clavier à vos projets de sorte que la carte UNO R3 puisse lire les entrées utilisateur faites avec le périphérique.

Les touches de claviers sont utilisées dans un grand nombre d'applications (téléphones / fax / fours / digicodes de portes etc…).

Pour ce projet nous allons utiliser un « matrix keypad ». C'est un clavier qui est conçu de telle sorte qu'il nécessite moins de pins de connexion qu'il a de touches. En effet, nous avons 17 touches (0-9, A-D, *, #) et 8 pins. Avec un clavier linéaire il faudrait 17 pins (une par touche) pour fonctionner. L'encodage par bit permet de combiner les pins pour en avoir besoin de beaucoup moins (ici deux fois moins)

Matériel nécessaire:

(2) x Elegoo Uno R3

(1) x "Membrane switch module"

(8) x Câbles mâle-mâle

Connection

Schéma de câblage

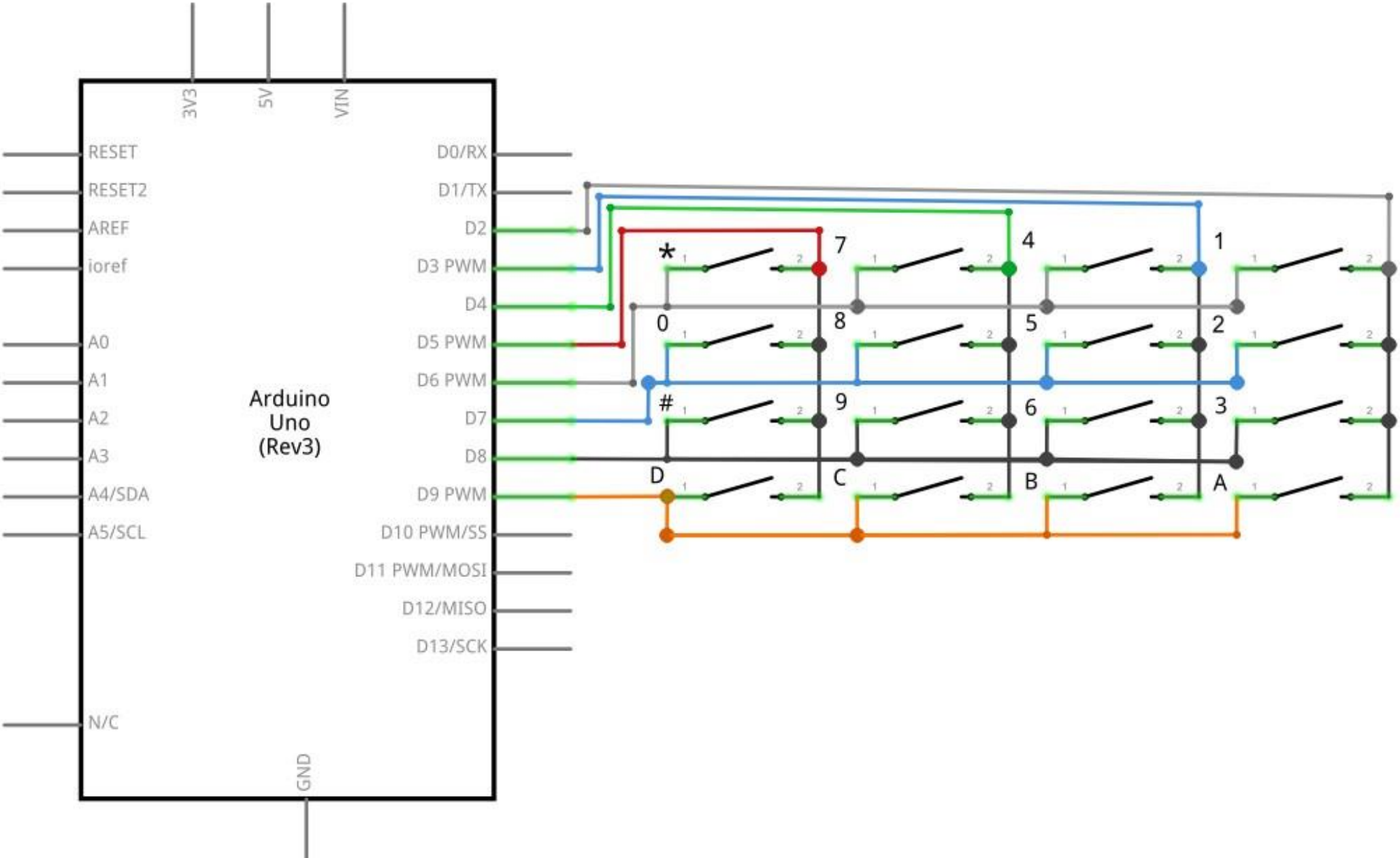
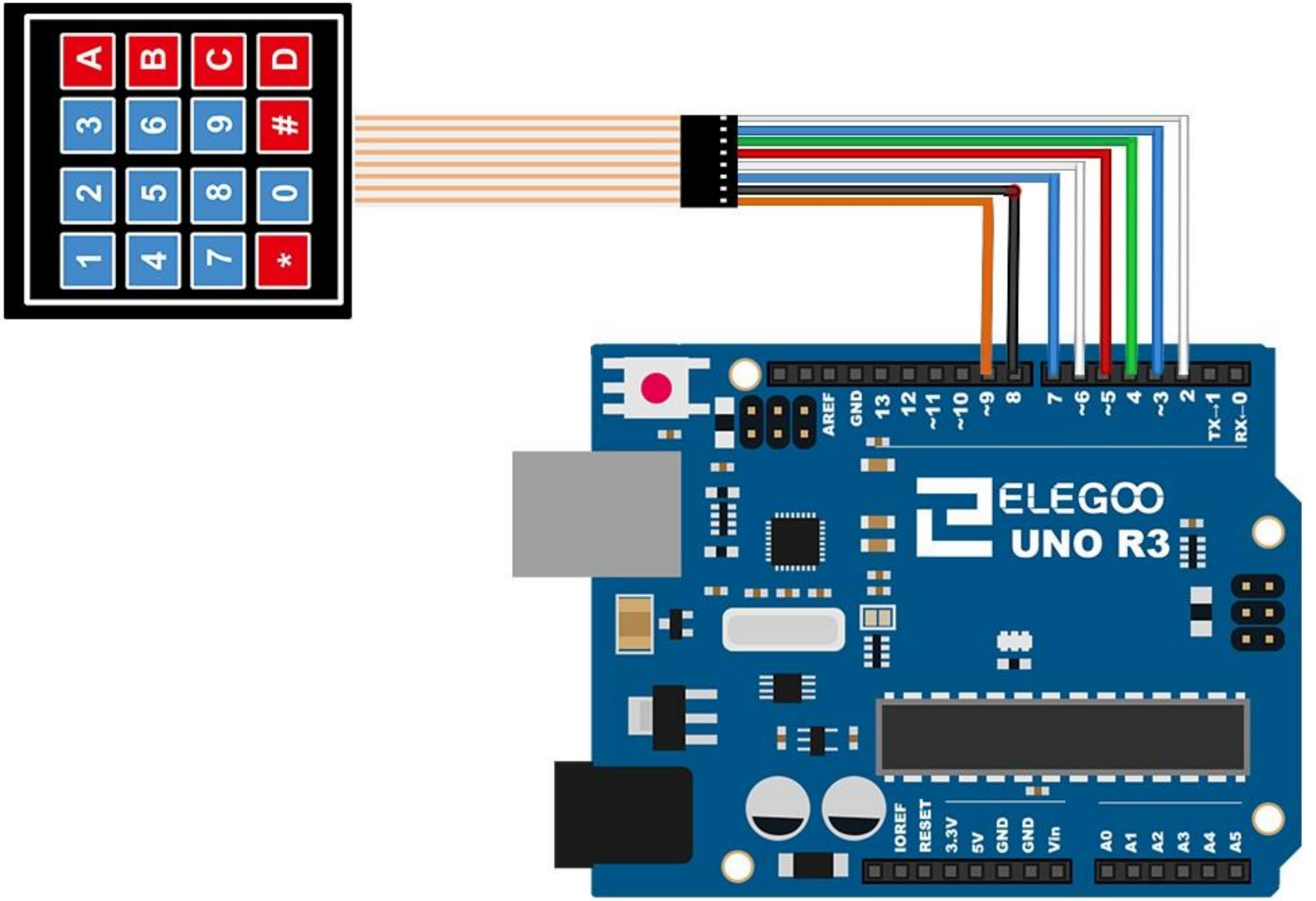


Diagramme de câblage



Pour plus de clarté, vous pouvez utiliser ce tableau pour être certain d'avoir réalisé le câblage comme il faut. Dans la colonne de gauche, le numéro de pin du clavier, dans la colonne de droite, la pin de la carte UNO R3

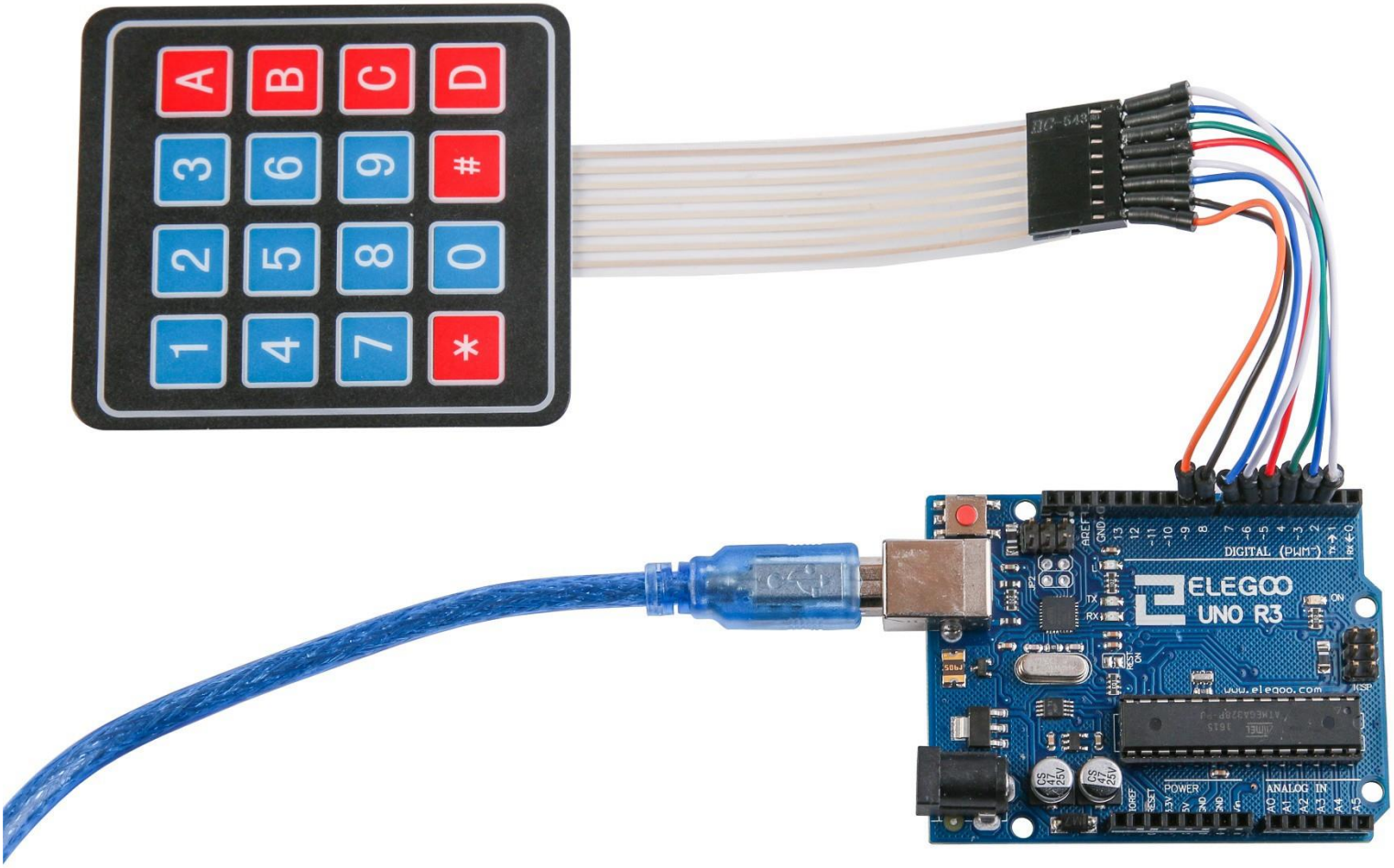
Keypad Pin	Connects to Arduino Pin...
1	D9
2	D8
3	D7
4	D6
5	D5
6	D4
7	D3
8	D2

Code

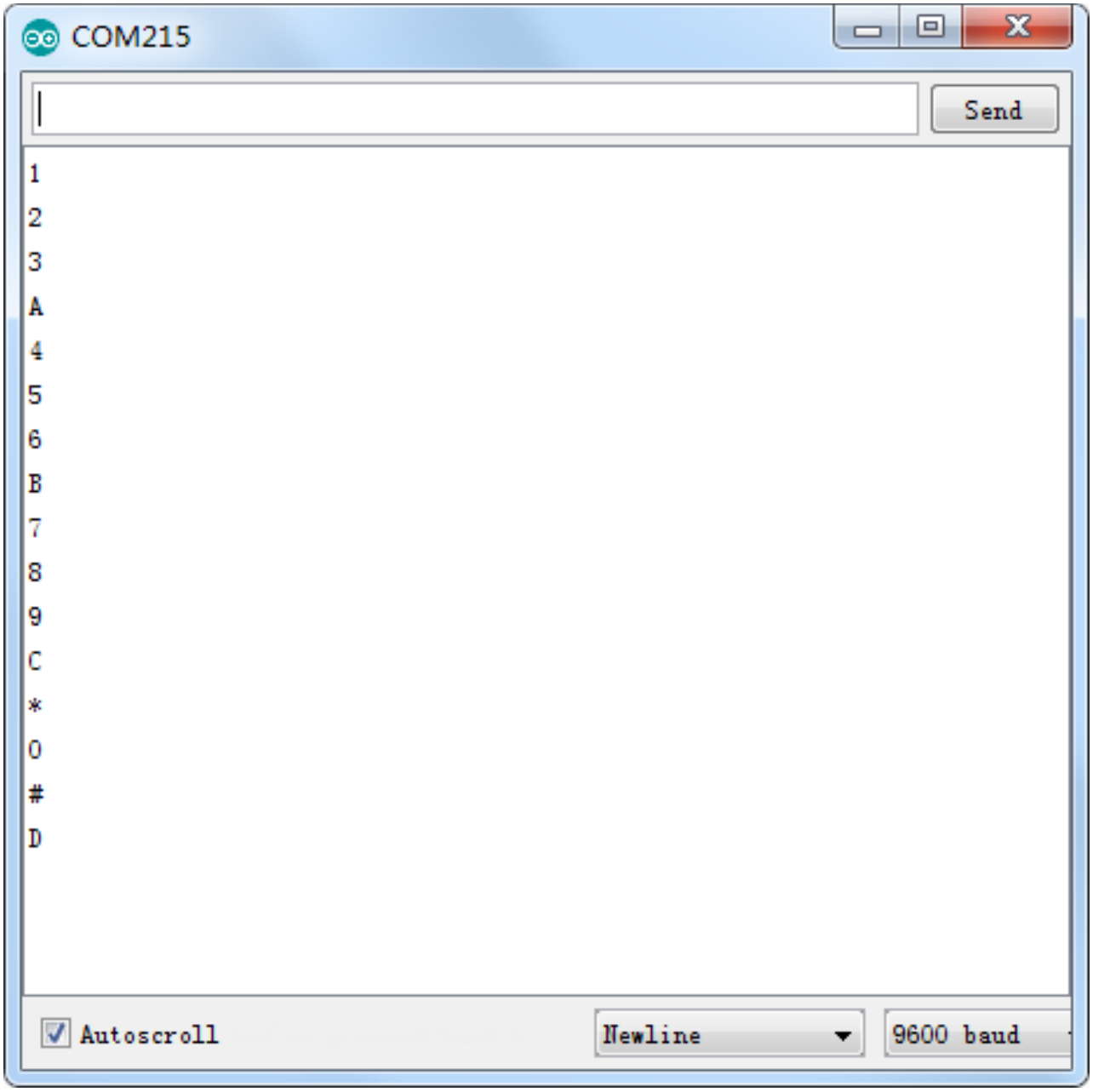
Après avoir réalisé le câblage, ouvrez le sketch "Leçon 11 Membrane Switch Module" et Téléversez-le sur la carte UNO R3 comme exposez à la leçon 2.

Avant toute chose, vérifiez bien d'avoir installé la bibliothèque < Keypad> qui gère le codage du clavier comme exposé à la leçon 1.

Illustration



Ouvrez ensuite le moniteur série sur votre ordinateur. Lorsque vous pressez une touche du petit clavier, vous pouvez observer la valeur correspondante s'afficher:



Leçon 12 DHT11 Temperature and Humidity Sensor

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser le capteur DHT11 qui permet une mesure de température et d'humidité.

C'est un capteur fiable qui permettra de répondre à la plupart des projets nécessitant un suivi de température et/ou d'humidité.

L'emploi d'une bibliothèque permet en plus une implémentation simple du code dans vos sketchs.

Matériel nécessaire:

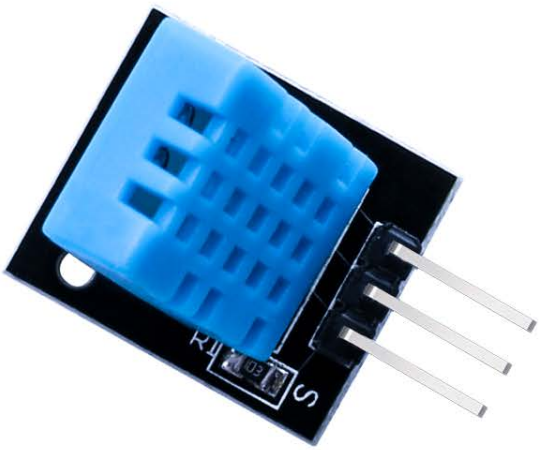
(1) x Elegoo Uno R3

(1) x DHT11 Temperature and Humidity module

(3) x Câbles Mâle-Femelle

Présentation du composant

Capteur DHT11:



Le DHT11 (Digitale Température Humidité 11) est un capteur qui génère un signal

digital en sortie codant une valeur de température et d'humidité mesurée en temps réel. Cette technologie est employée car elle permet de produire une mesure fiable sur le long terme. Il contient en plus de ces capteurs un microcontrôleur 8-bit.
Encore une fois, il est utilisé dans un grand nombre d'applications qui nécessitent de moduler le fonctionnement d'un appareil en fonction de conditions de température ou d'humidité.

Paramètres:Humidité
relative: Résolution:
16Bit Répétabilité:
±1% RH
Précision: At 25℃ ±5% RH
Temps de réponse: 1 / e (63%) of 25℃ 6s
1m / s air 6s
Hystérésis: <± 0.3% RH
Stabilité: <± 0.5% RH / an
Température: Résolution:
16Bit
Répétabilité: ±0.2℃
Gamme de températures: At 25℃ ±2℃
Temps de réponse: 1 / e (63%) 10S
Alimentation: DC 3.5～5.5V
Courant : 0.3mA standby 60μA
Etalonnage: 2 secondes

Connection

Schéma de câblage

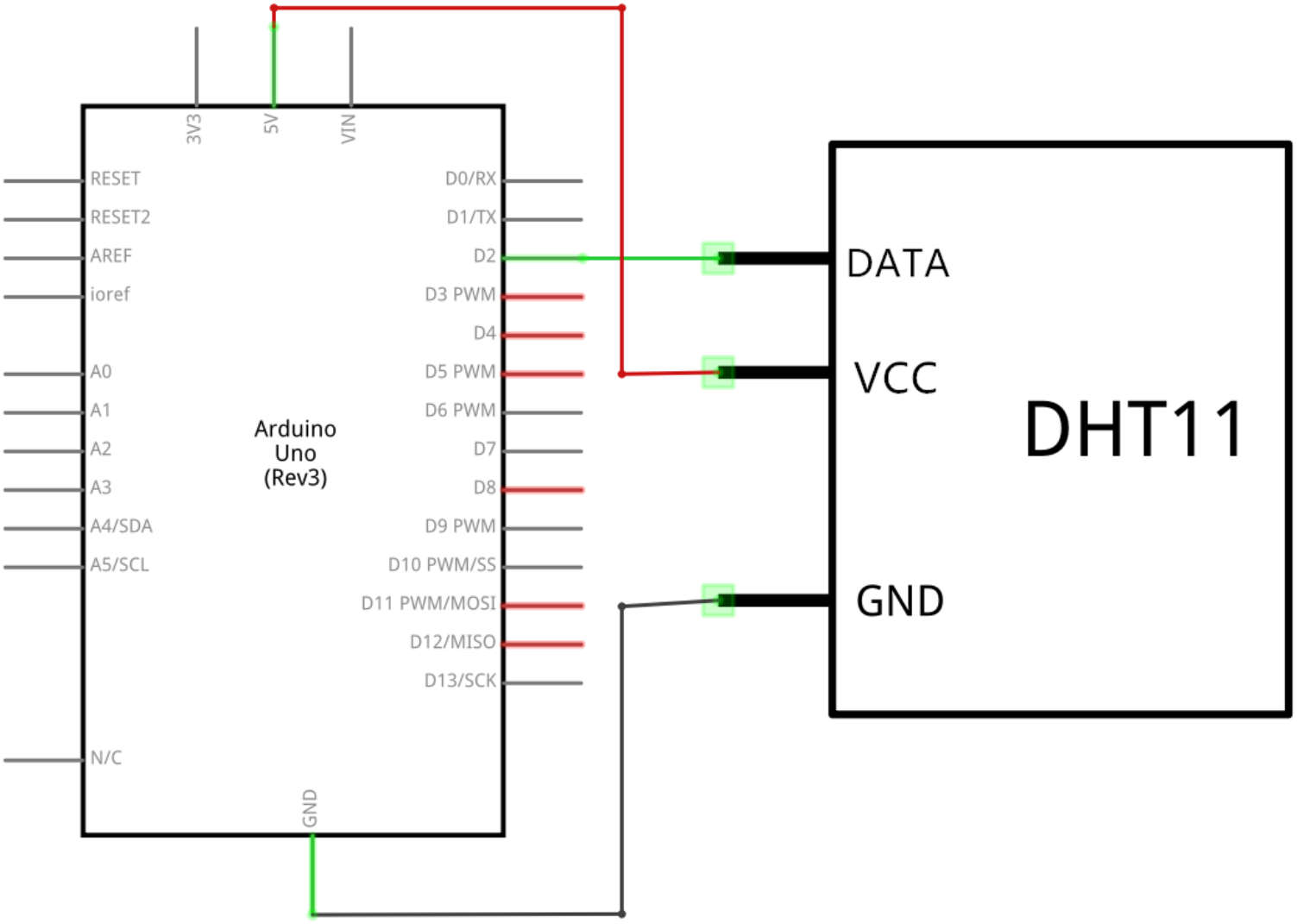
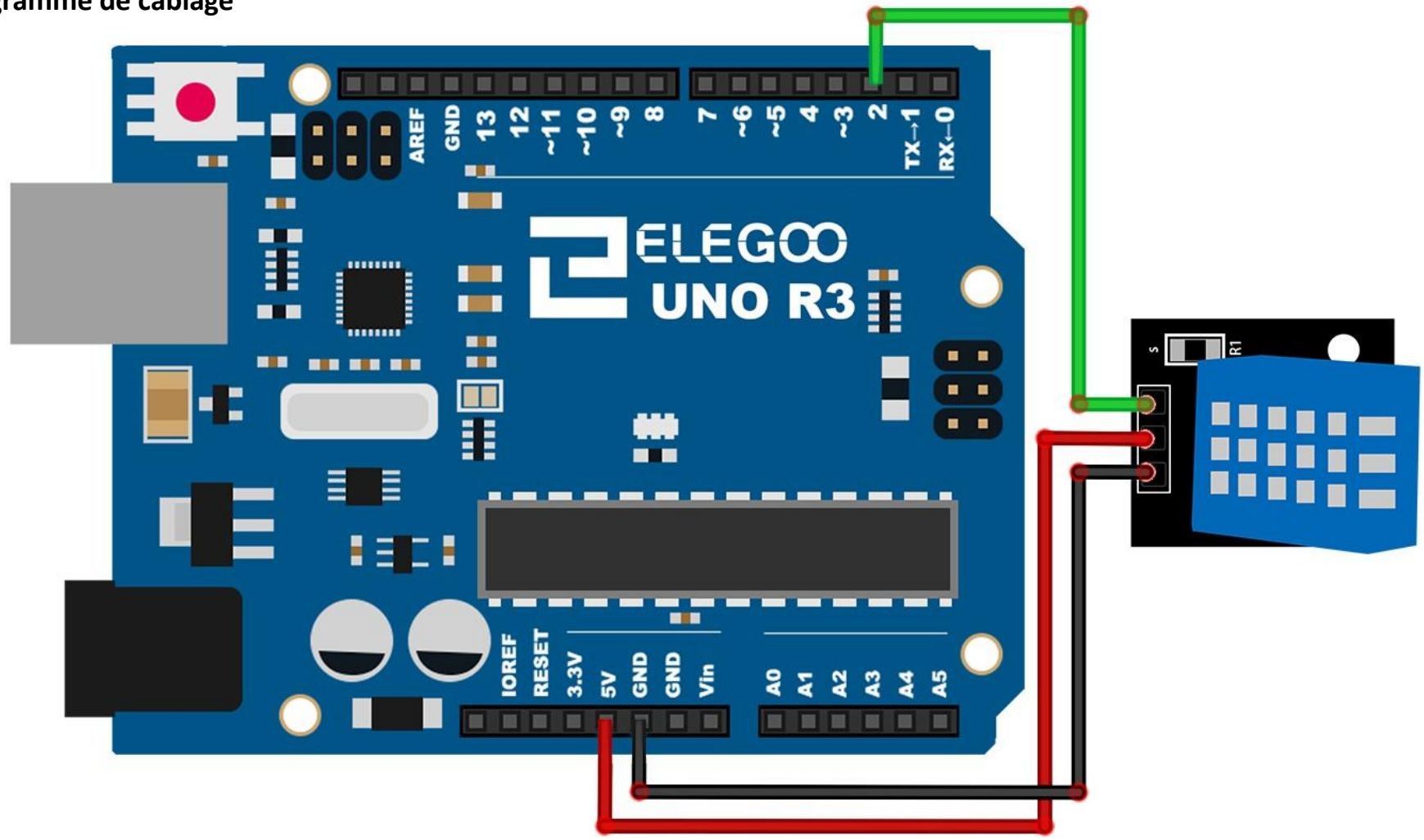


Diagramme de câblage

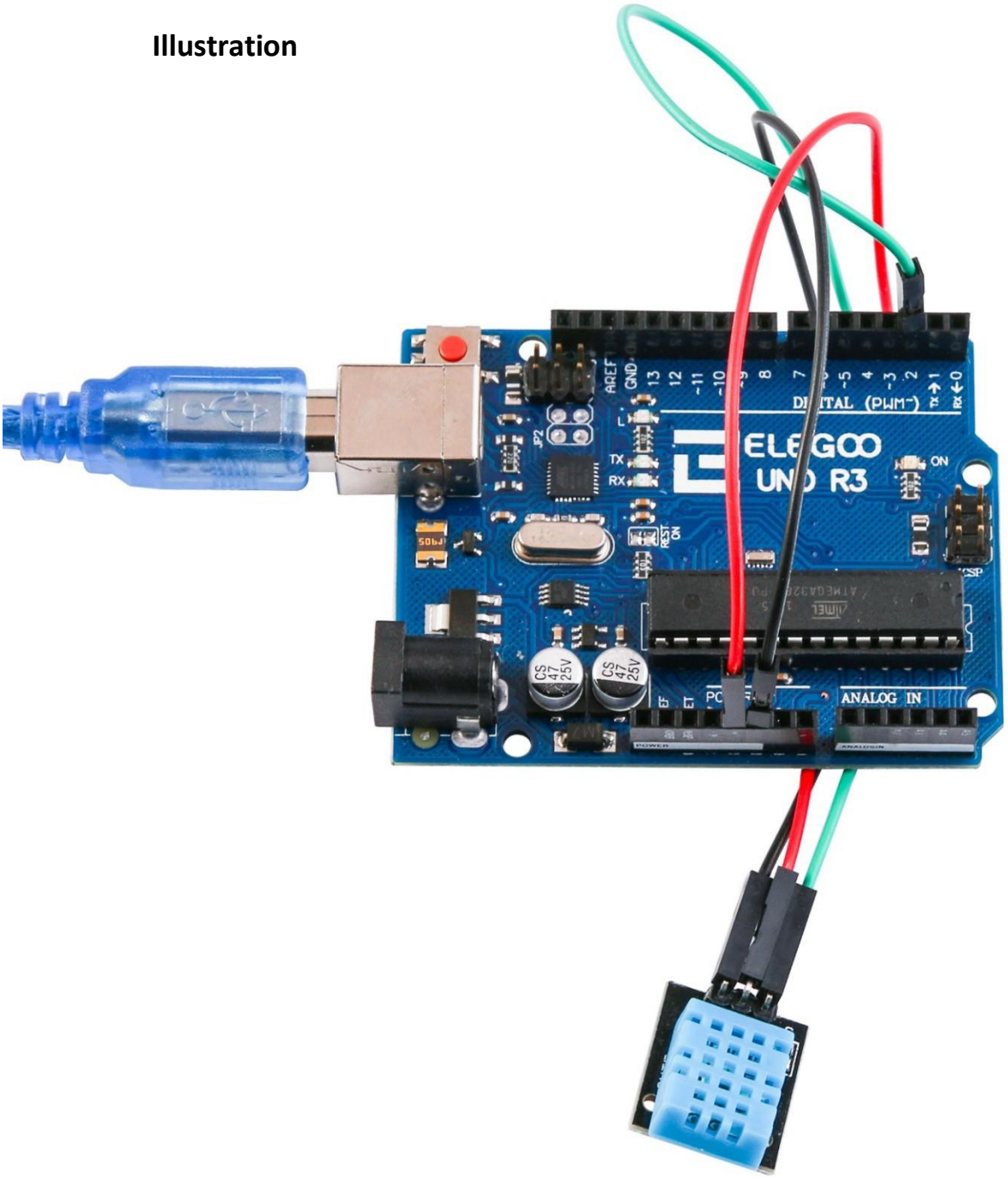


Code

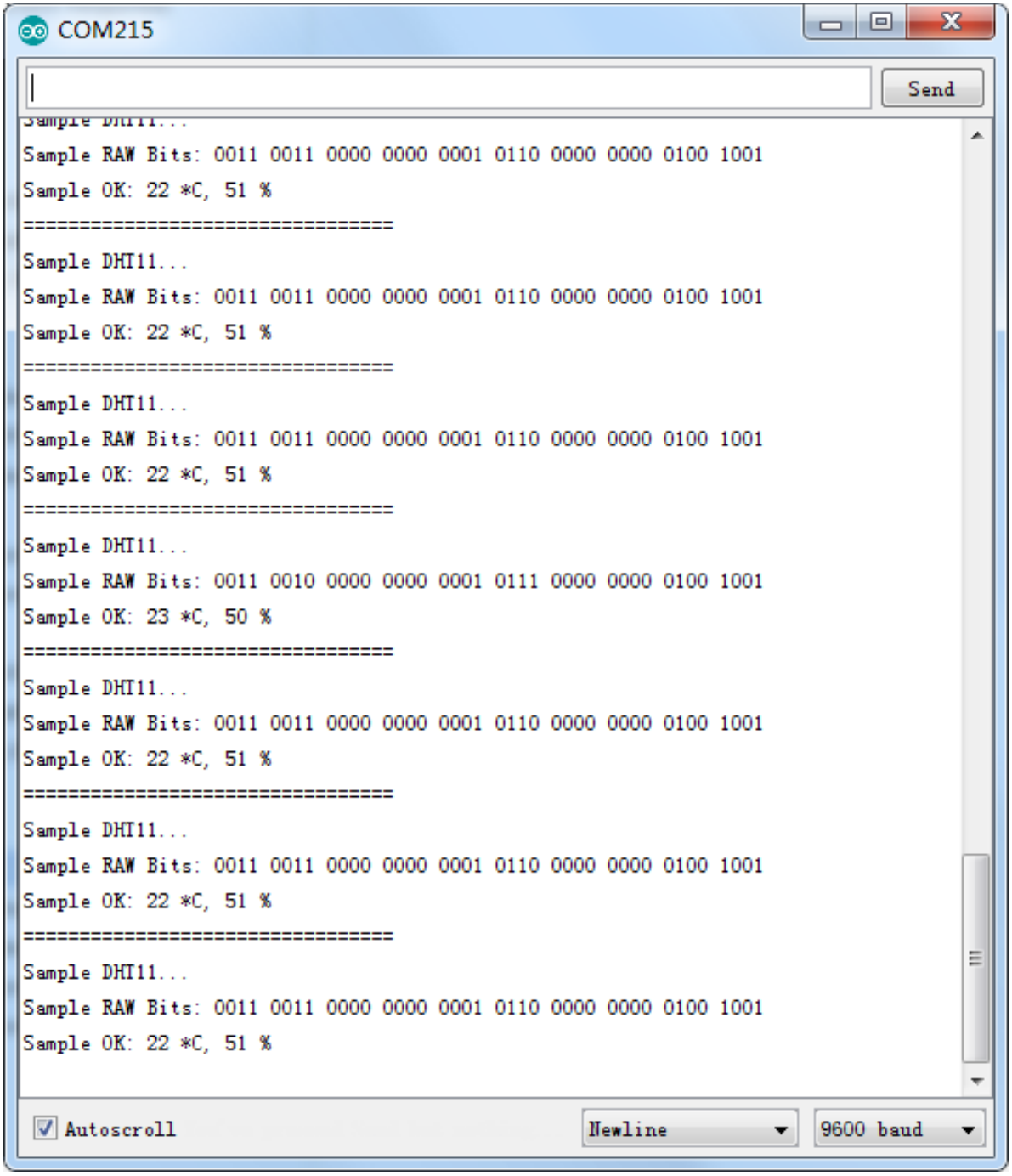
Ouvrez le sketch “Leçon 12 DHT11 Temperature and Humidity Sensor” et Téléversez-le sur la carte UNO R3.

Avant toute chose, veillez à avoir installé la bibliothèque <SimpleDHT> comme exposé à la leçon 1.

Illustration



Ouvrez le moniteur série sur votre ordinateur, les mesures s'affichent.



Leçon 13 Analog Joystick Module

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser un joystick analogique du même type que ceux que vous pouvez retrouver sur les manettes des PS2.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Joystick module

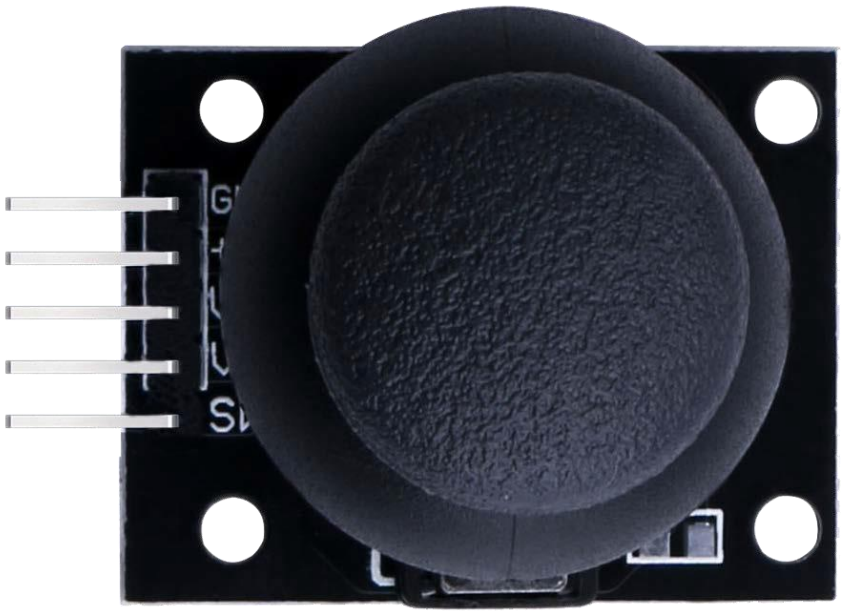
(5) x Câbles Mâle-Femelle

Présentation du composant

Joystick

Le module a 5 pins: VCC, Ground, X, Y, Key. Les noms affectés aux pins peuvent légèrement varier en fonction de la provenance des modules. Le joystick est analogique, ce qui permet une lecture précise de la position et offre beaucoup plus de possibilité qu'un simple joystick tactile. En plus, il est possible d'appuyer sur le joystick pour faire une fonction « select ».

Pour la lecture de la position X/Y du joystick nous utiliserons des entrées analogiques de la carte UNO R3 et pour lire le bouton, une entrée digitale.



Connection

Schéma de câblage

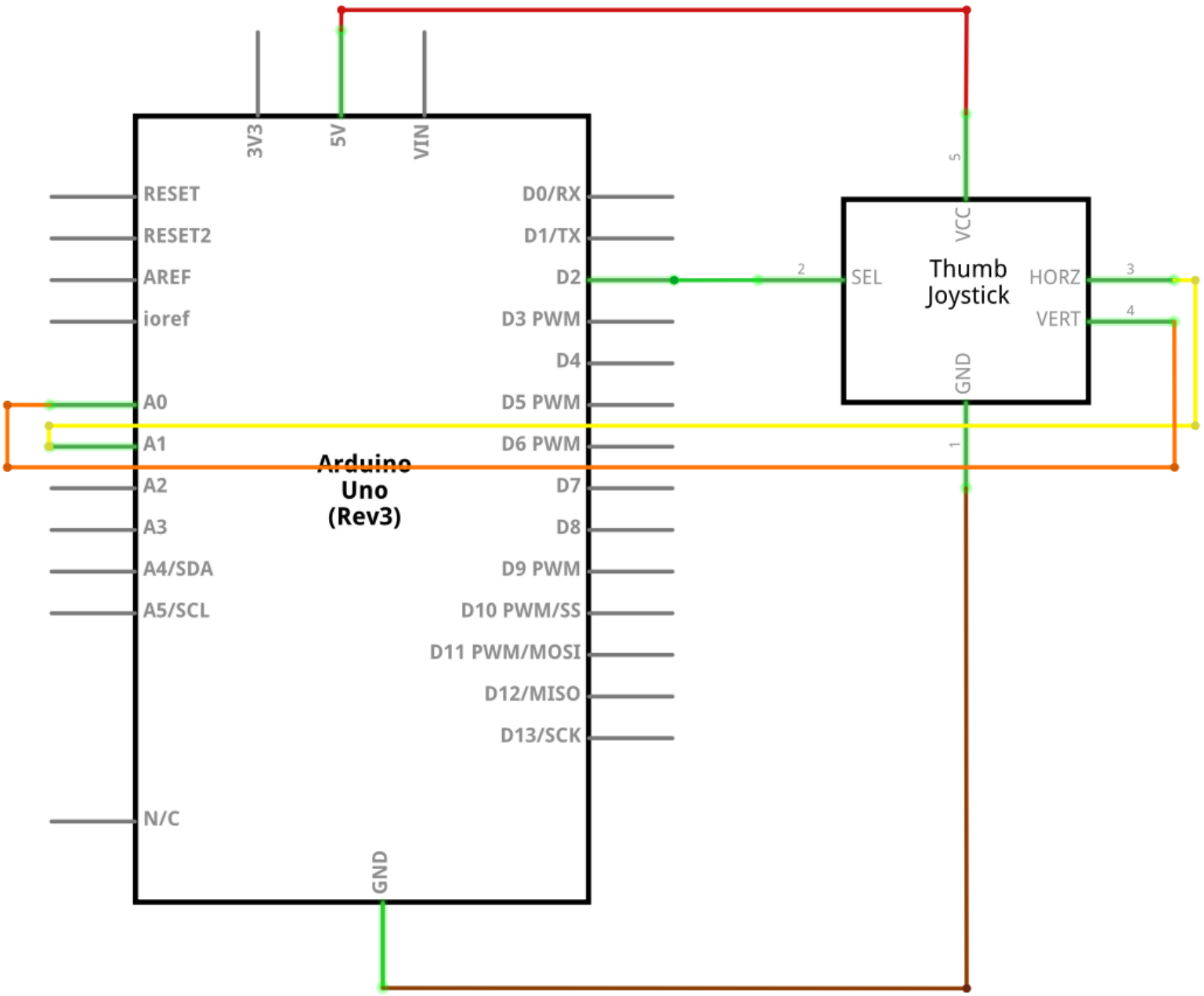
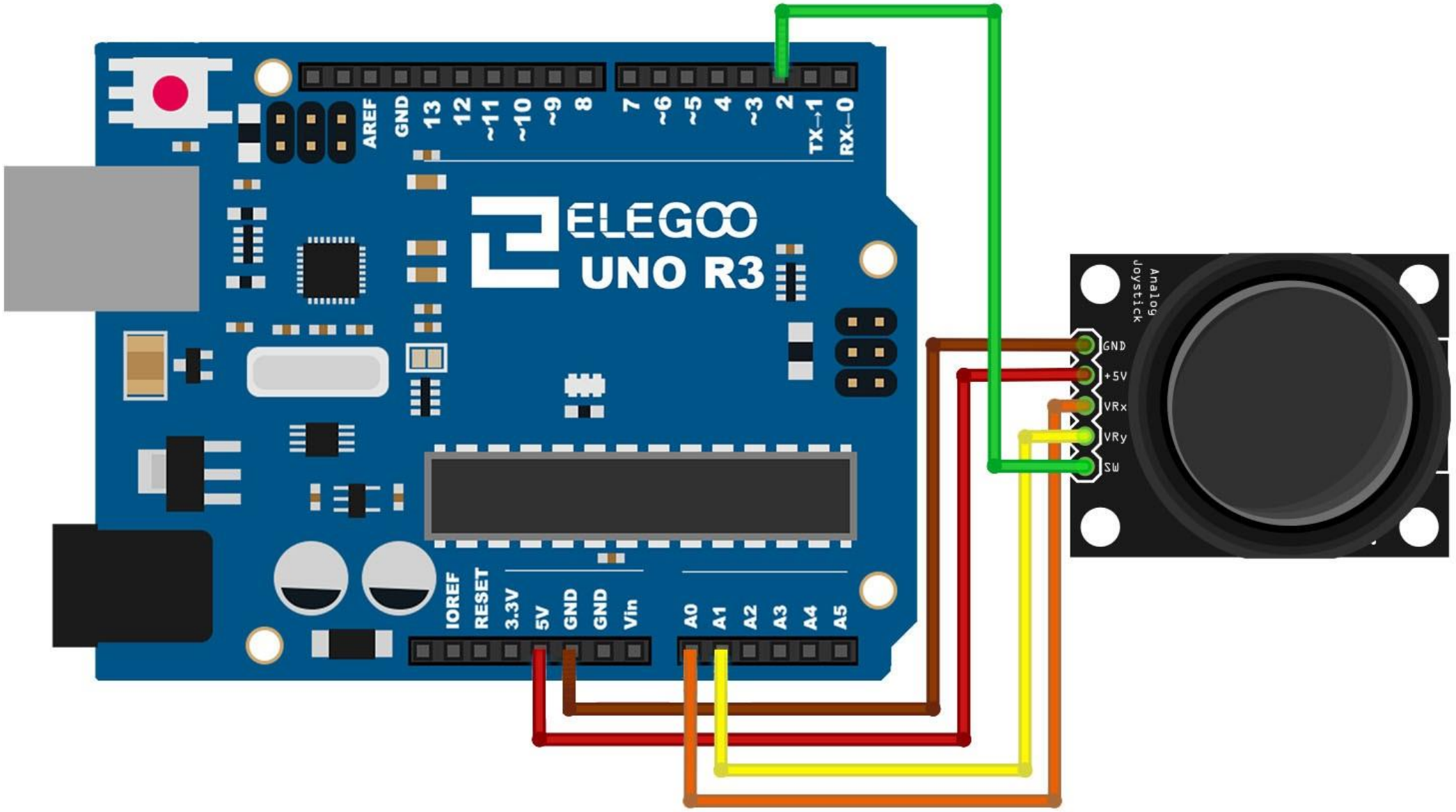


Diagramme de câblage



5 connexions sont donc nécessaires pour le joystick.

Les connexions sont : Key, Y, X, Voltage and Ground.

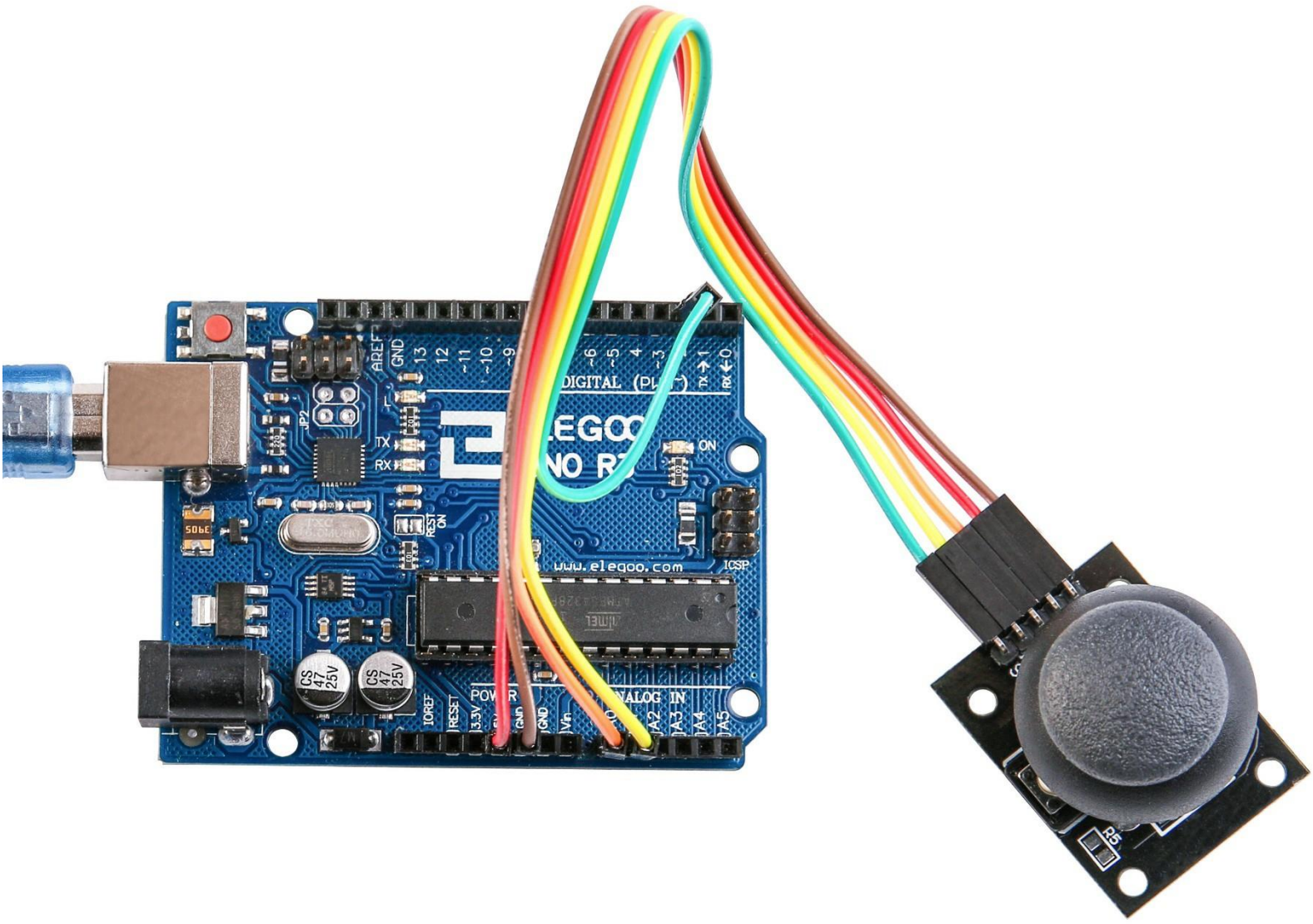
“Y et X” sont analogiques and “Key” est digitale. Le branchement de la pin pour la fonction « select » n’est pas obligatoire si non nécessaire au projet.

Code

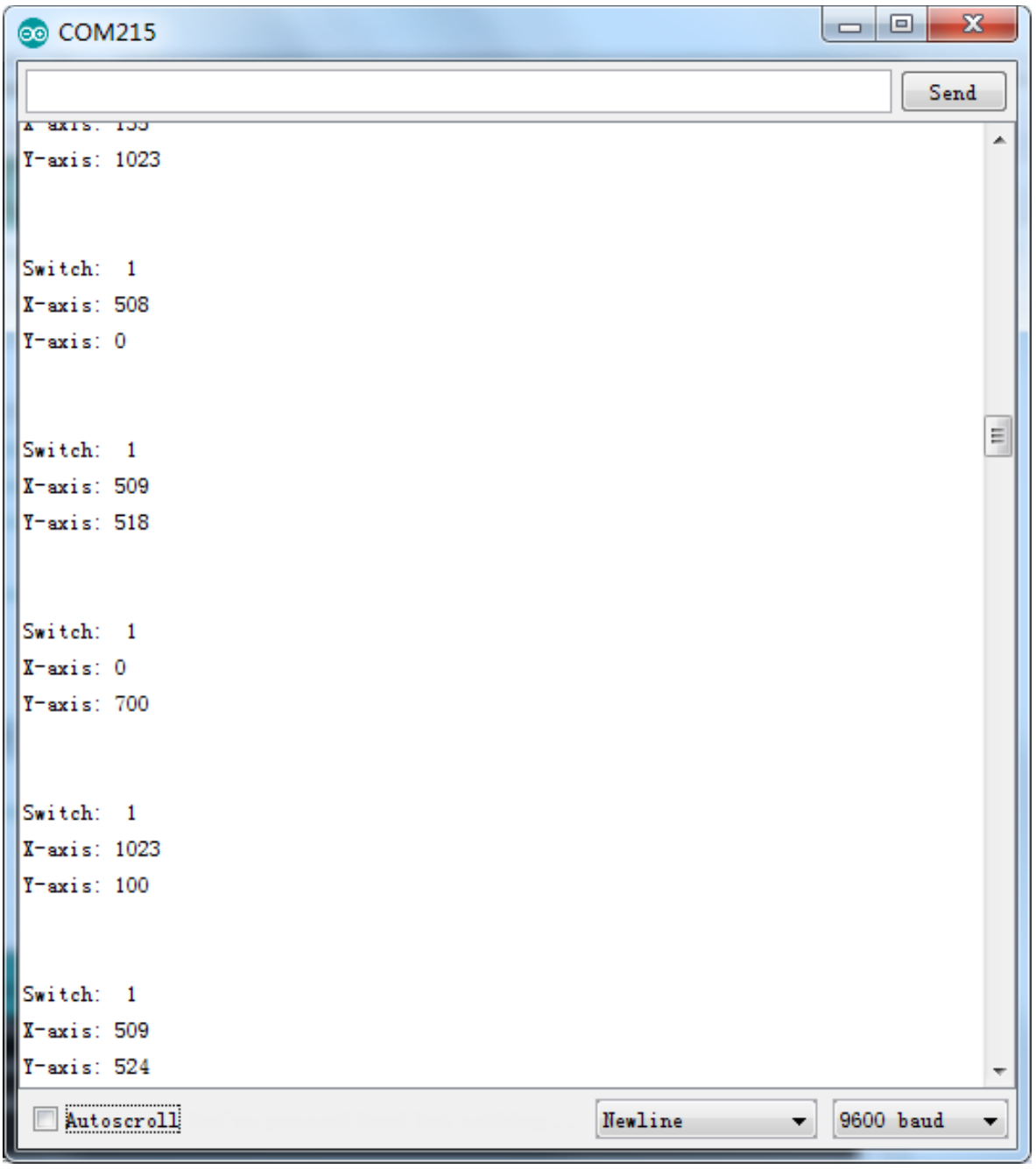
Après avoir réalisé le câblage, ouvrez le sketch “Leçon 13 Analog Joystick Module” et Téléversez-le sur la carte.

Le joystick fonctionne comme un potentiomètre. Il va retourner une valeur entre 0 et 1024, sachant qu’en position centrale il renvoie 512.

Illustration



Ouvrez le moniteur série pour observer les valeurs:



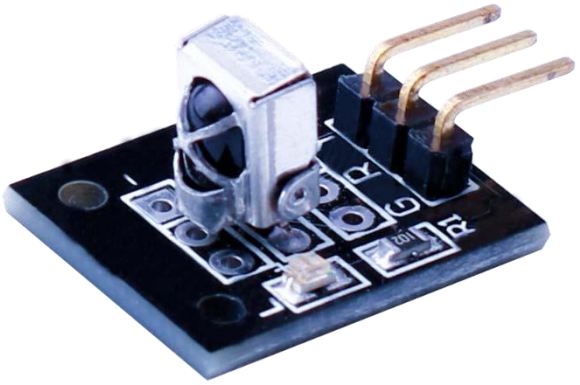
Leçon 14 IR Receiver Module

But de la leçon

L'utilisation de l'infrarouge est un très bon moyen de réaliser un contrôle à distance. Les télécommandes infrarouges ne sont pas compliquées à mettre en oeuvre.
Dans cette leçon, vous allez apprendre à brancher un récepteur infrarouge à la carte UNO R3 et utiliser une bibliothèque dédiée à ce capteur.
Dans le sketch, vous aurez tout le code hexadécimal disponible avec la télécommande fournie et vous allez pouvoir détecter la touche appuyée sur celle-ci.

Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x Capteur IR
(1) x Télécommande IR
(3) x Câbles Mâle-Femelle



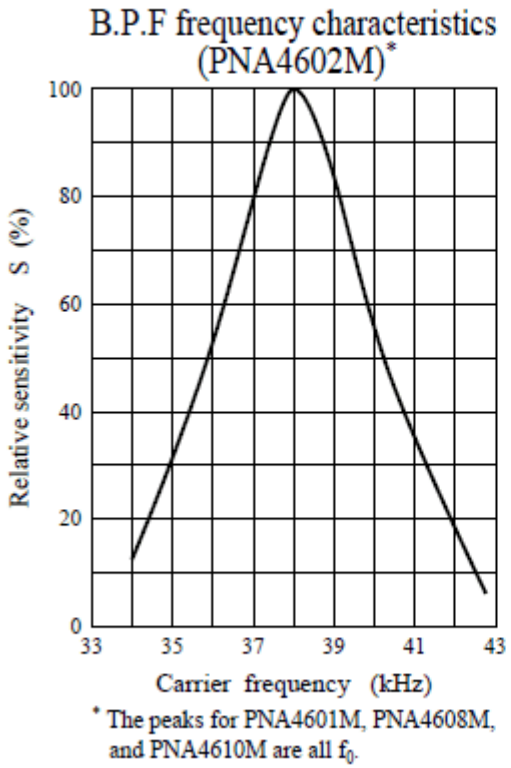
Présentation du composant

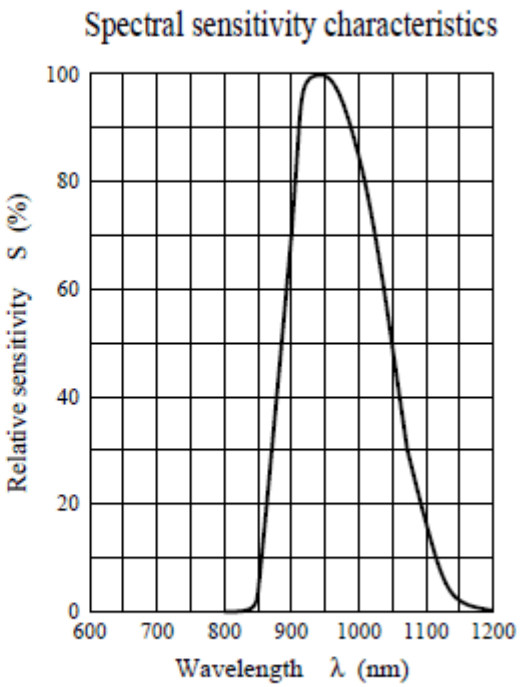
Capteur IR:

C'est un circuit équipé d'une cellule photosensible réglée sur les infrarouges. On les utilise beaucoup pour faire des récepteurs de télécommandes de TV ou lecteur DVD.
Nota : la lumière infrarouge n'est pas visible de l'oeil humain.

Les capteurs IR sont équipés d'un démodulateur intégré qui travaille sur des signaux de 38kHz.
Un émetteur IR doit clignoter pour être détecté par le capteur, le simple fait d'être allumé ne permet pas de détection.

Qu’est-il possible de mesurer?





Comme vous pouvez le voir sur ces graphiques, le pic de fréquence est à 38kHz et le pic de la LED est à 940nm. Il est possible de travailler entre 35kHz et 41kHz, mais la sensibilité va chuter. De la même manière, vous pouvez utiliser une LED allant de 850 à 1100 nm, mais cela ne marchera pas aussi bien qu’avec celle de 850nm.

Connection

Schéma de câblage

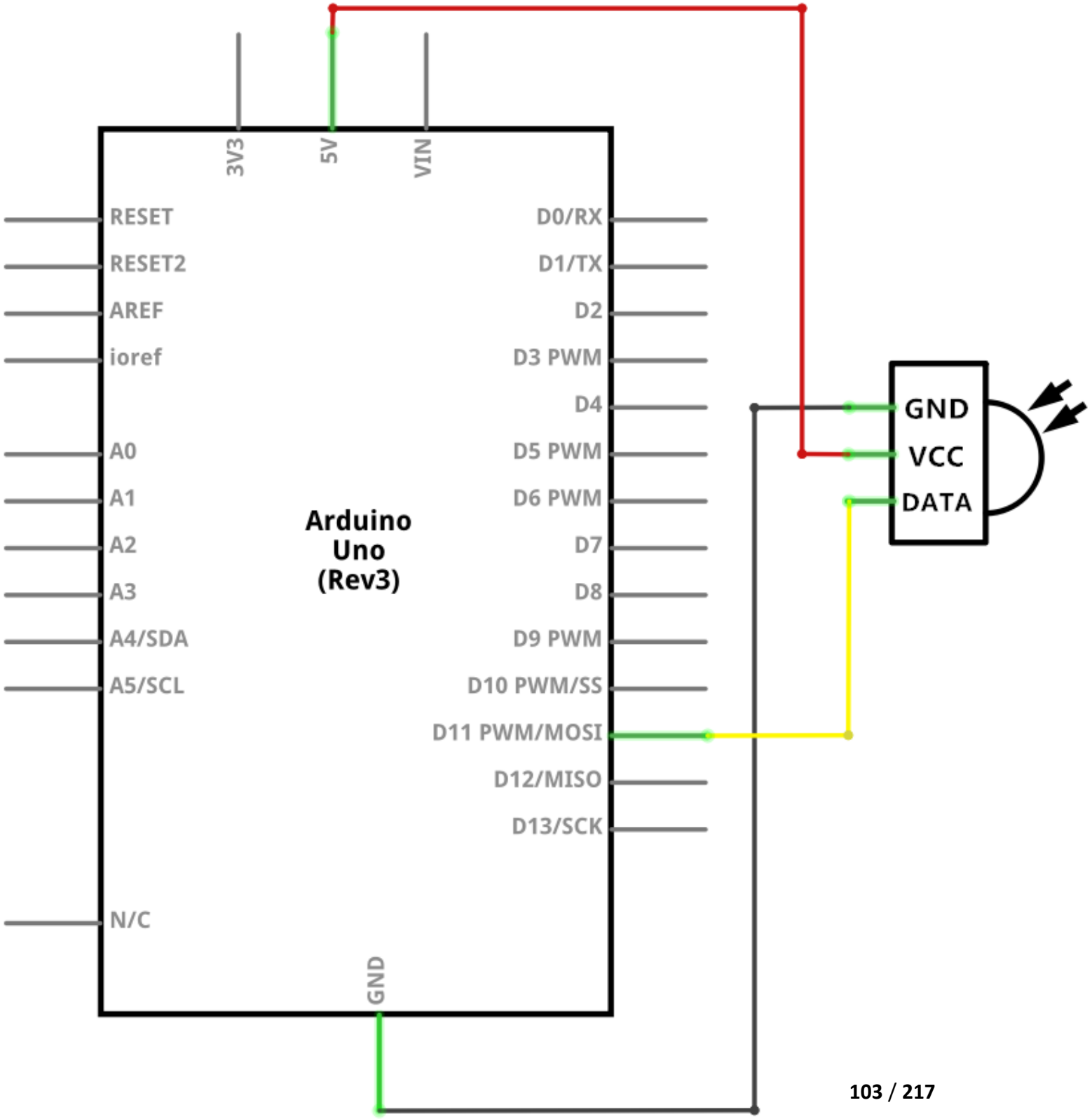
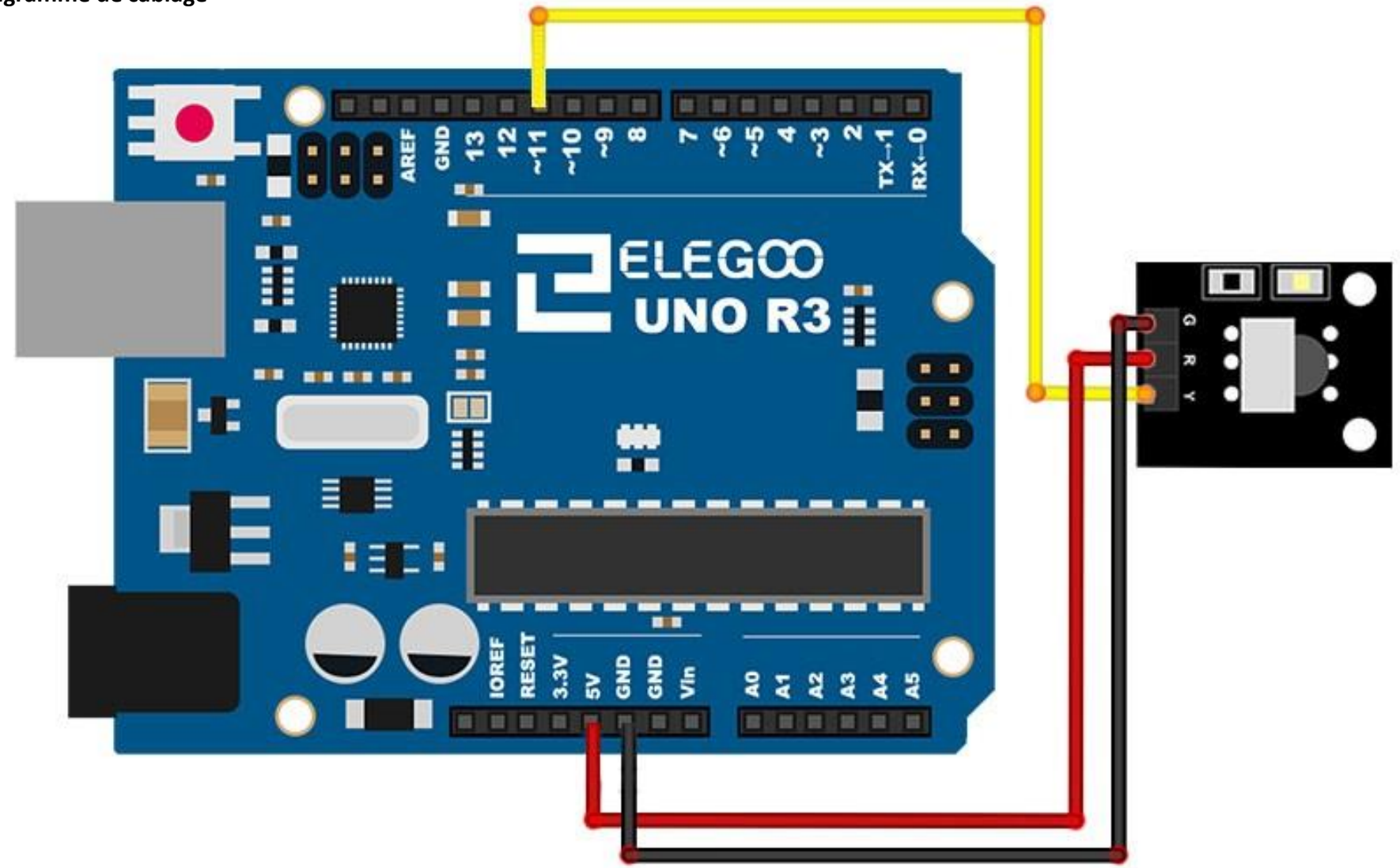


Diagramme de câblage



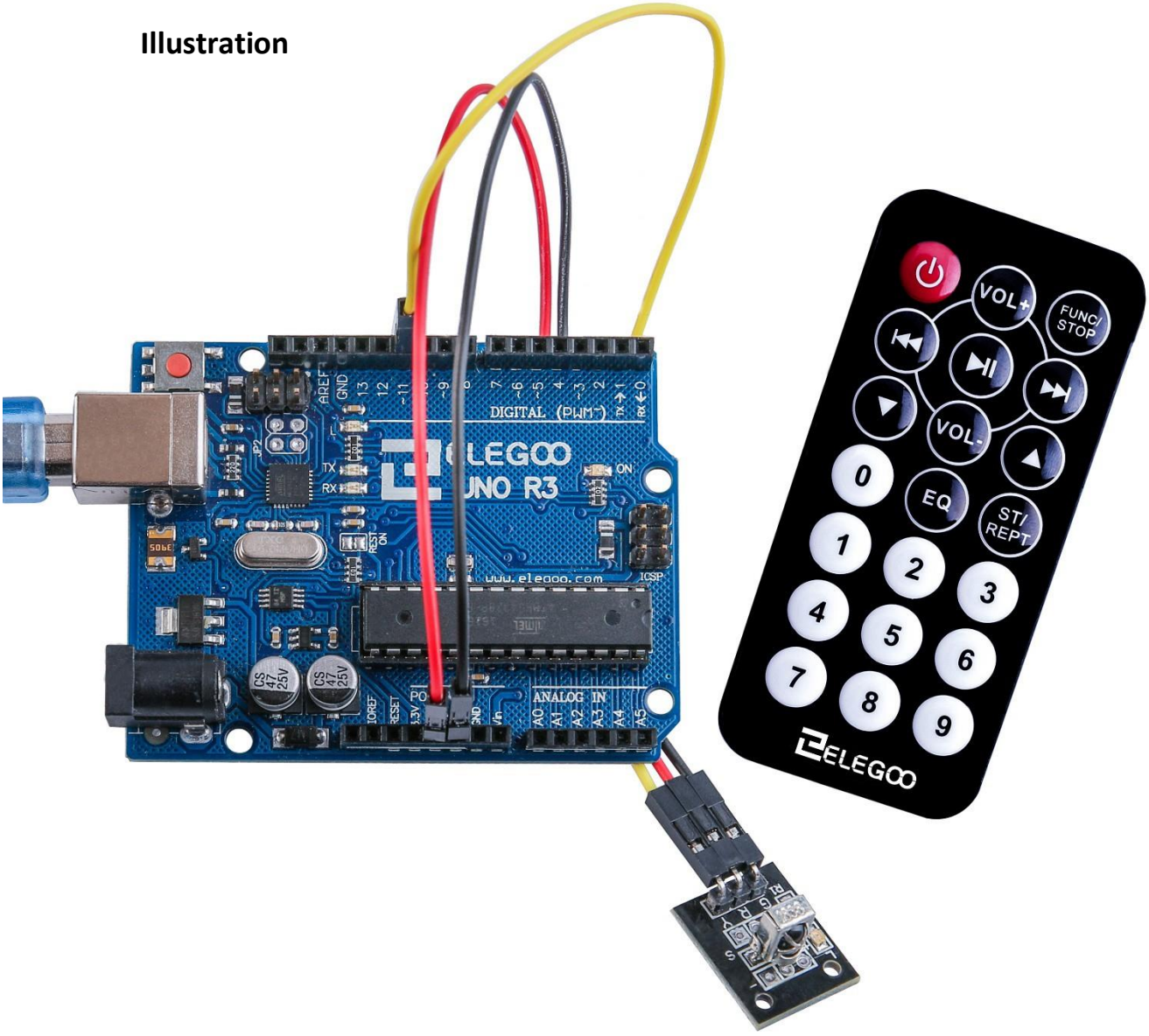
Il y a 3 pins sur le capteur :
+Vcc, Masse, Signal

Code

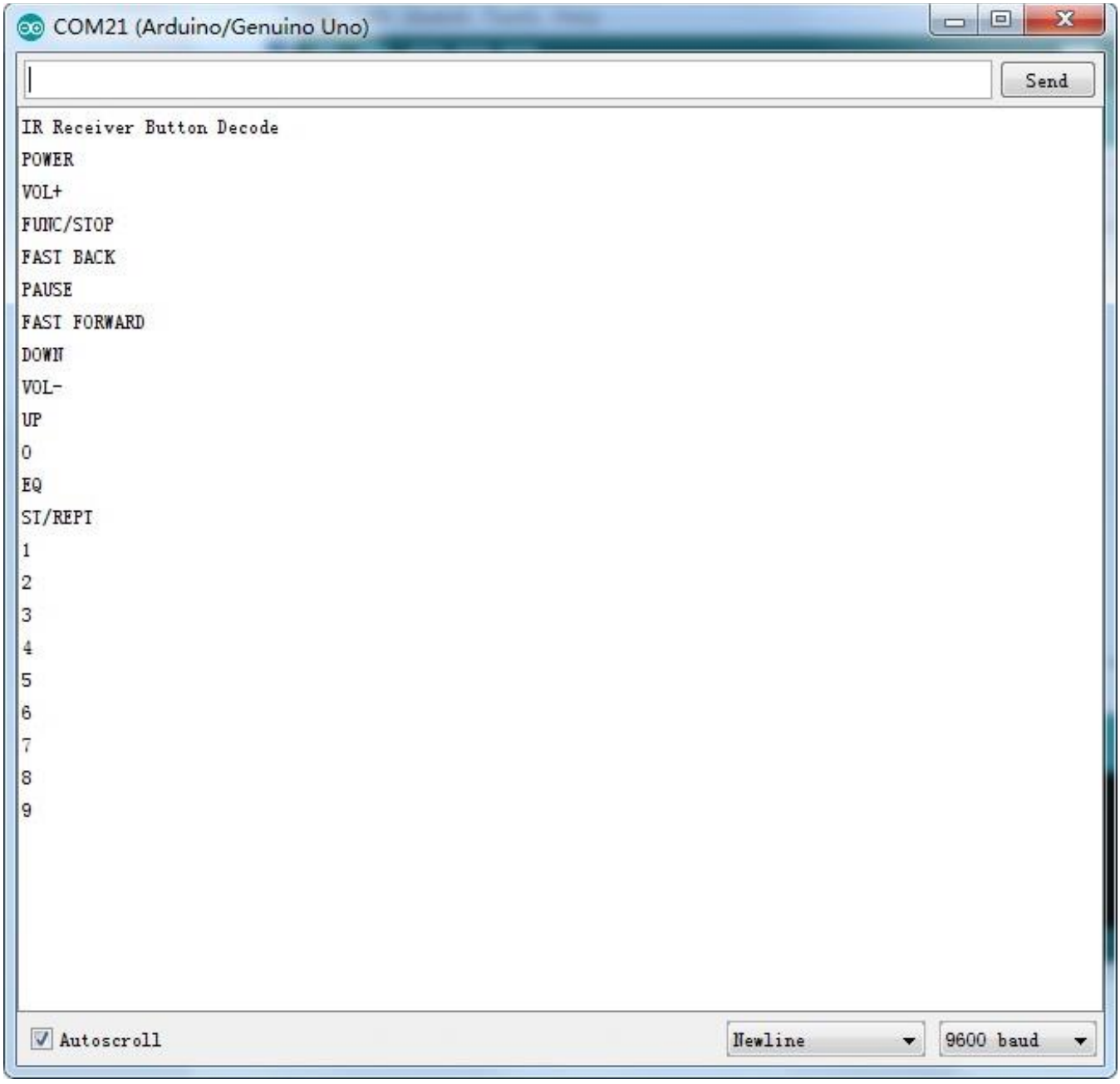
Après avoir réalisé le câblage, ouvrez le sketch “Leçon 14 IR Receiver Module” et Téléversez-le sur la carte UNO R3.
Avant toute chose, vérifiez bien l’installation de la bibliothèque < IRremote > comme expliqué à la leçon 1.

Ensuite, nous devons retirer le fichier <RobotIRremote> du répertoire pour éviter des conflits lors de l’utilisation.

Illustration



Ouvrez le moniteur série, vous pouvez observer ce qu'il se produit en cas d'appui sur la télécommande.



Leçon 15 MAX7219 LED Dot Matrix Module

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser un afficheur 64 leds de type MAX7219.

Depuis que ces modules utilisent des contrôleurs, ils n'ont besoin que de 3 pins sur la carte UNO.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Module Max7219

(5) x Câbles Mâle-Femelle

Présentation du composant

MAX7219 LED Dot Matrix Module

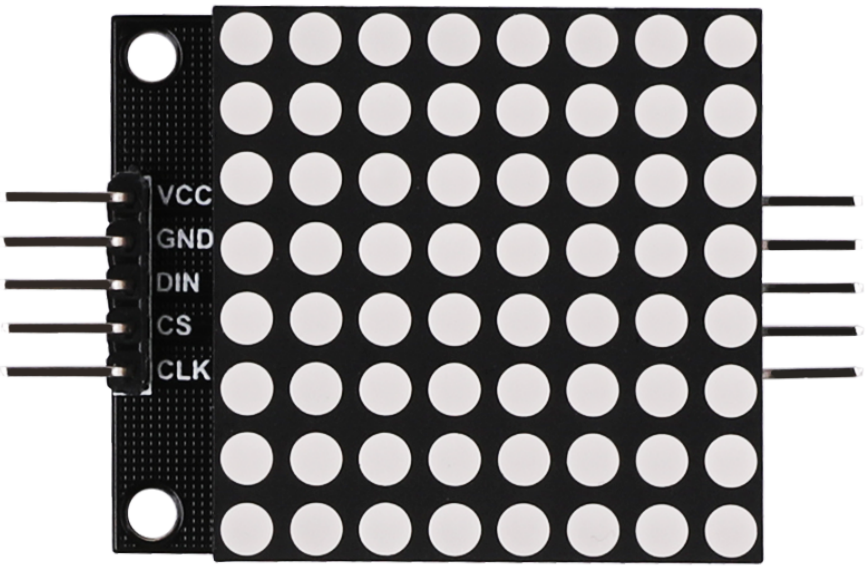
Ce projet est en fait une carte UNO montée en série avec une matrice de 8x8 LEDS. Le circuit intégré MAX7219 est un driver à cathode commune qui peut piloter des afficheurs 7 segments ou des barres de LEDS (jusqu'à 64).

Spécifications:

Voltage: DC 4.7V – 5.3V

Courant: 320mA

Courant maximum: 2A



Connection

Schéma de câblage

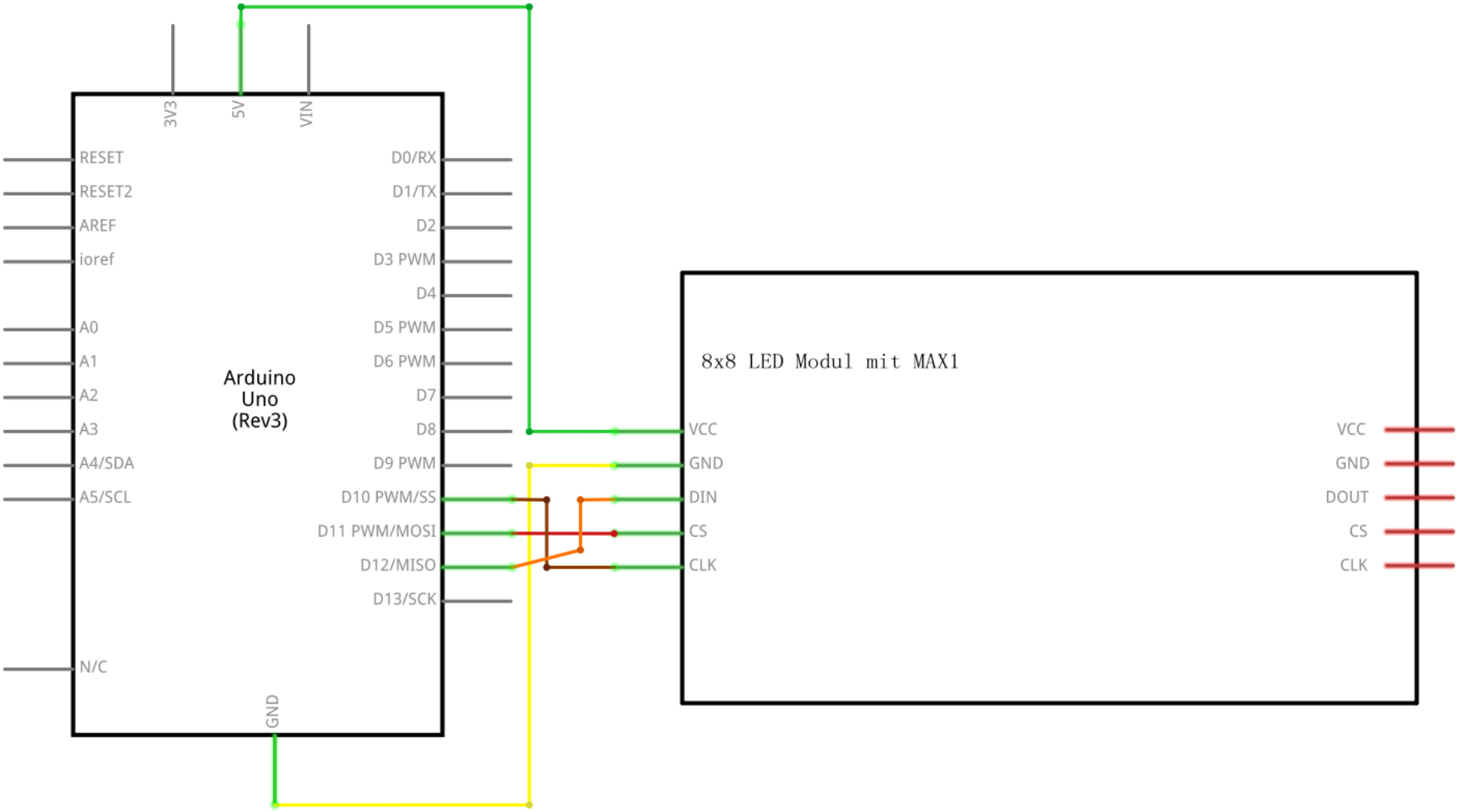
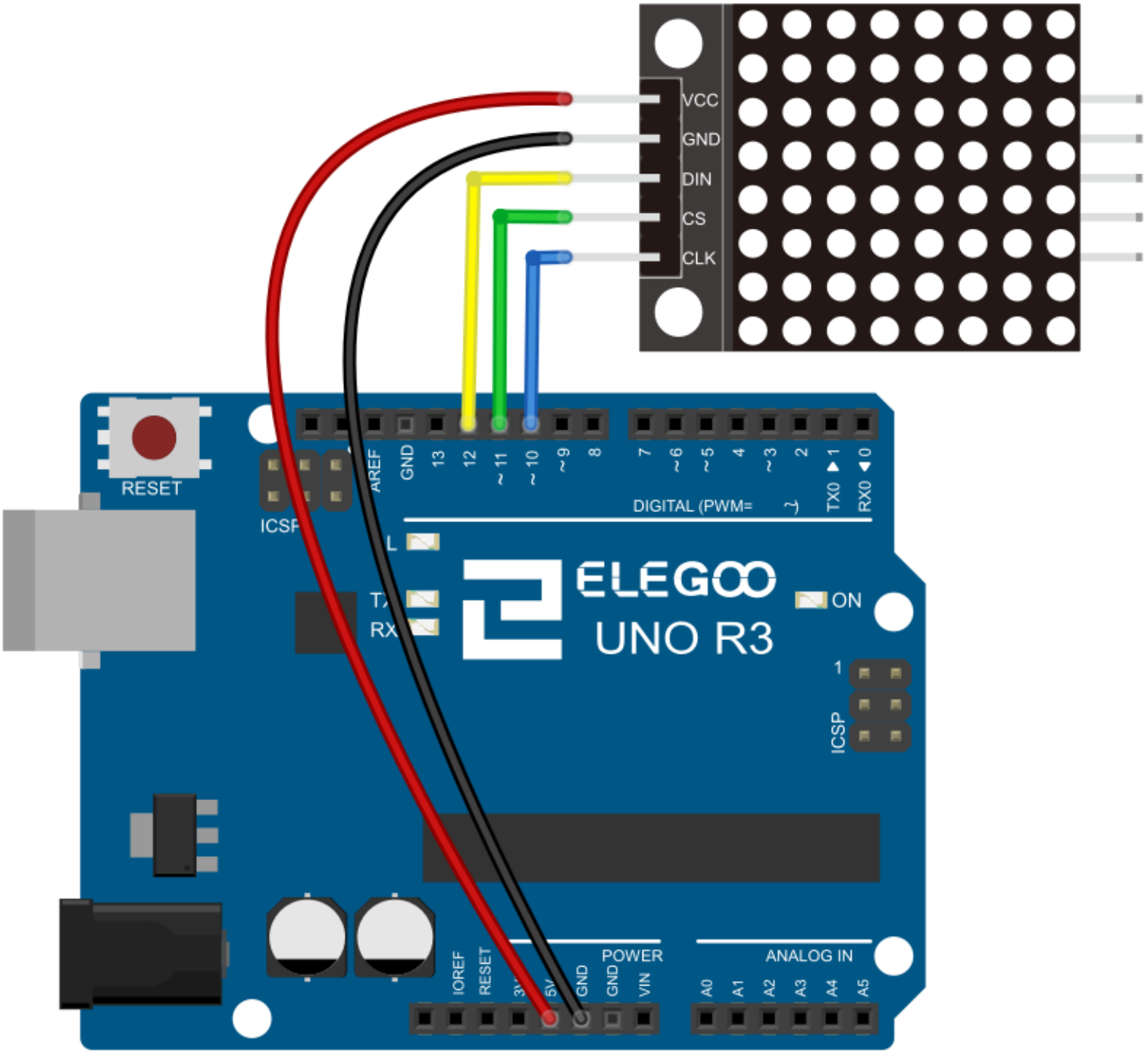


Diagramme de câblage



Pin 12 : DIN / Pin 11 : CS / Pin 10 CLK.

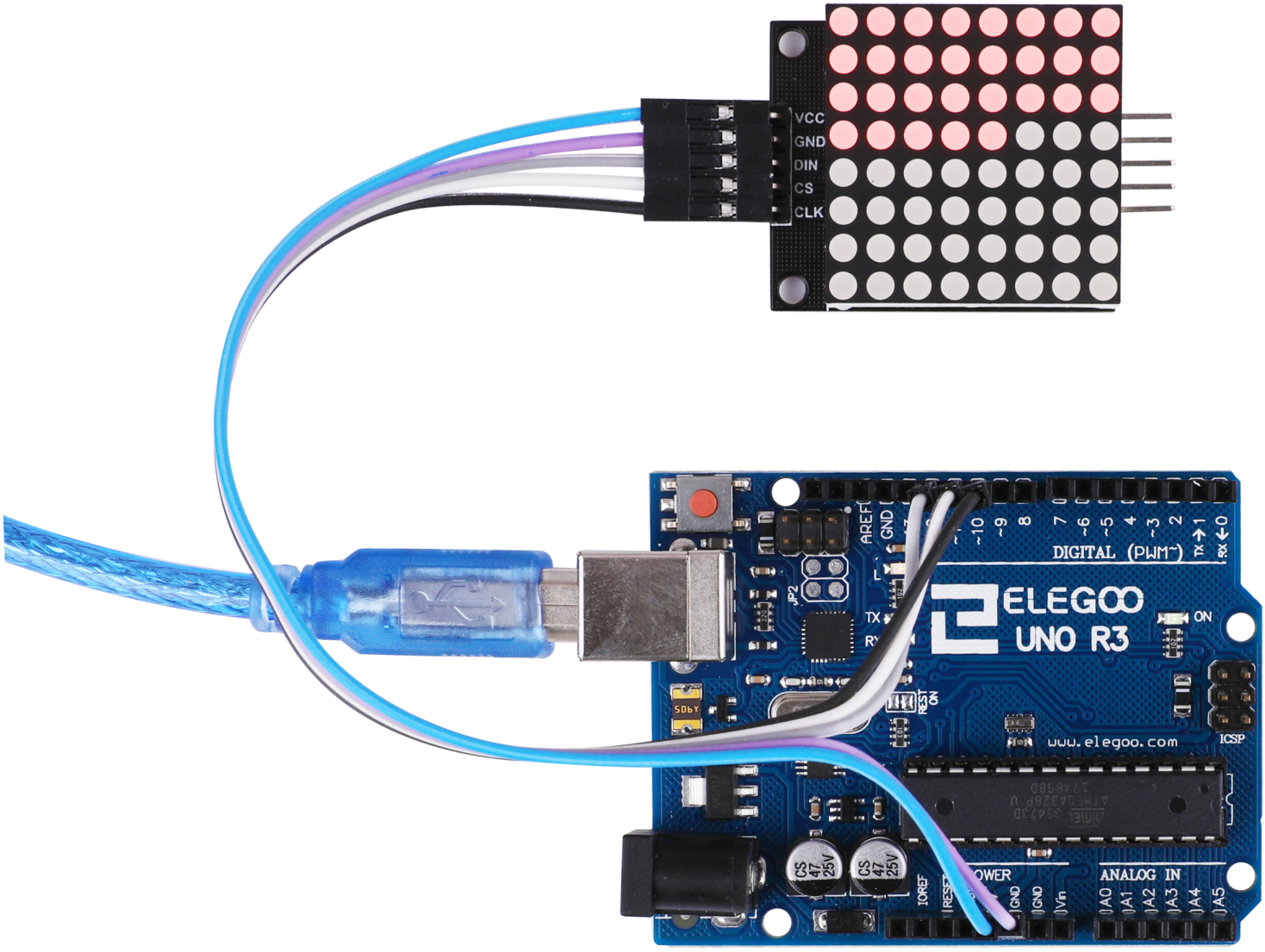
Code

Le sketch utilise la bibliothèque “Maxmatrix”.

Après avoir réalisé le câblage, ouvrez le sketch ”Leçon 15 MAX7219 LED Dot Matrix Module » et téléversez-le sur la carte UNO R3.

Avant toute chose, veillez à bien avoir installé la bibliothèque < LedControl > comme exposé à la leçon 1.

Illustration



Leçon 16 GY-521 Module

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser un module de mesure d'inertie. Ce capteur est notamment utilisé dans les robots à deux roues qui tiennent en équilibre de manière autonome.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Module GY-521

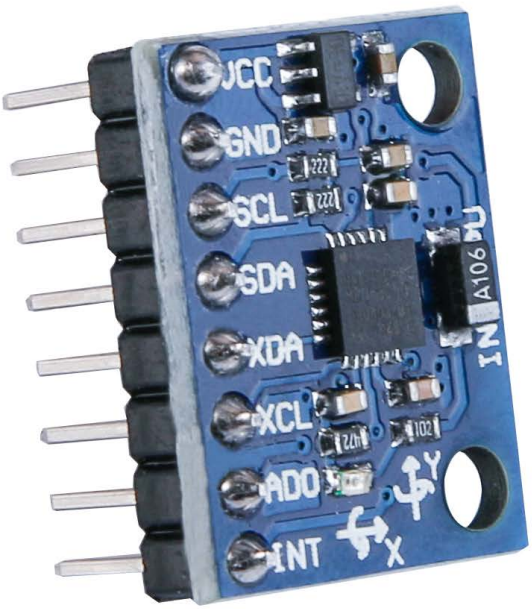
(4) x Câbles Mâle-Femelle

Présentation du composant

CAPTEUR GY-521

Ce capteur contient un gyroscope et un accéléromètre intégrés. C'est un composant

ɔé

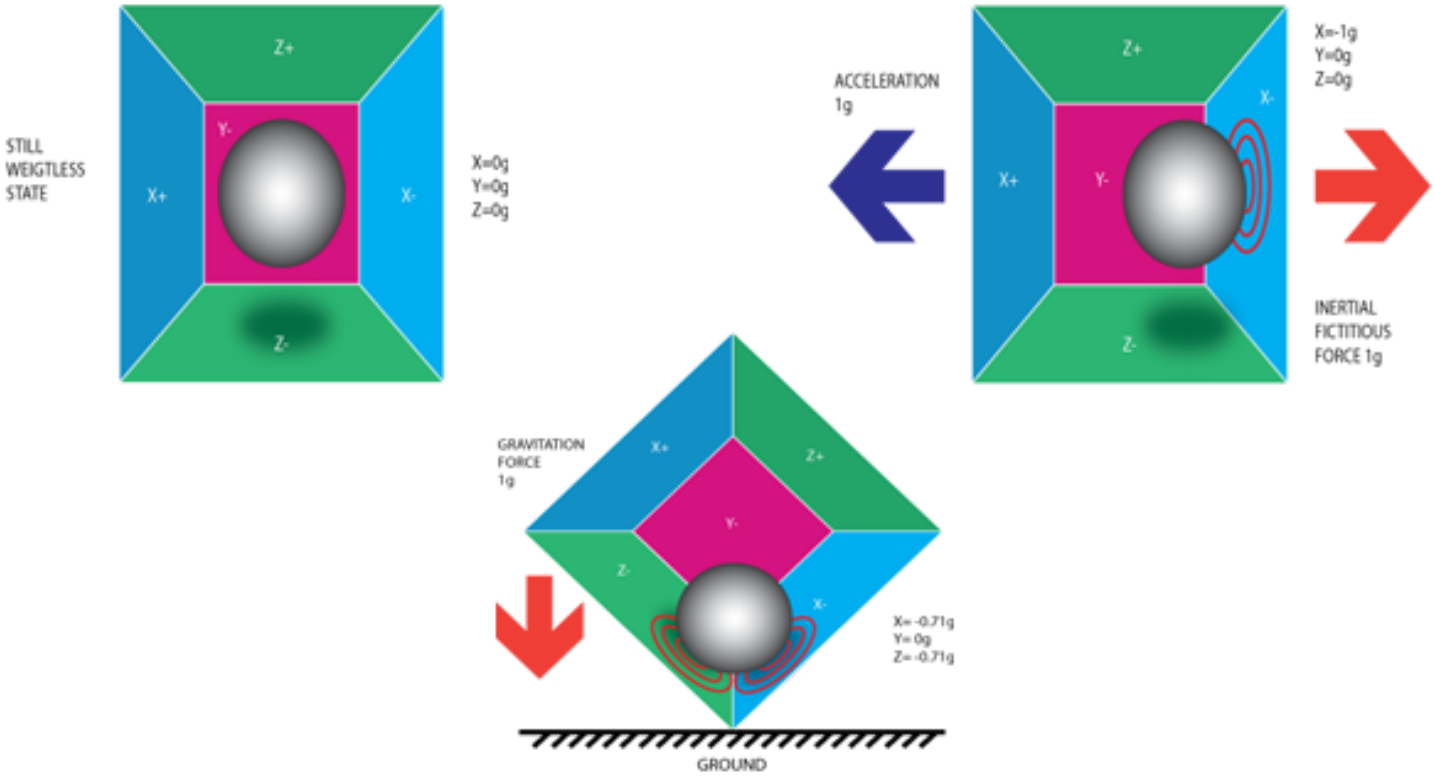


Ce capteur est un incontournable dans toute sortes de gadgets. On le retrouve dans les smartphones, habits, les manettes de jeux etc…
Le capteur permet de connaitre l'évolution dans l'espace d'un objet.

Comment cela fonctionne?

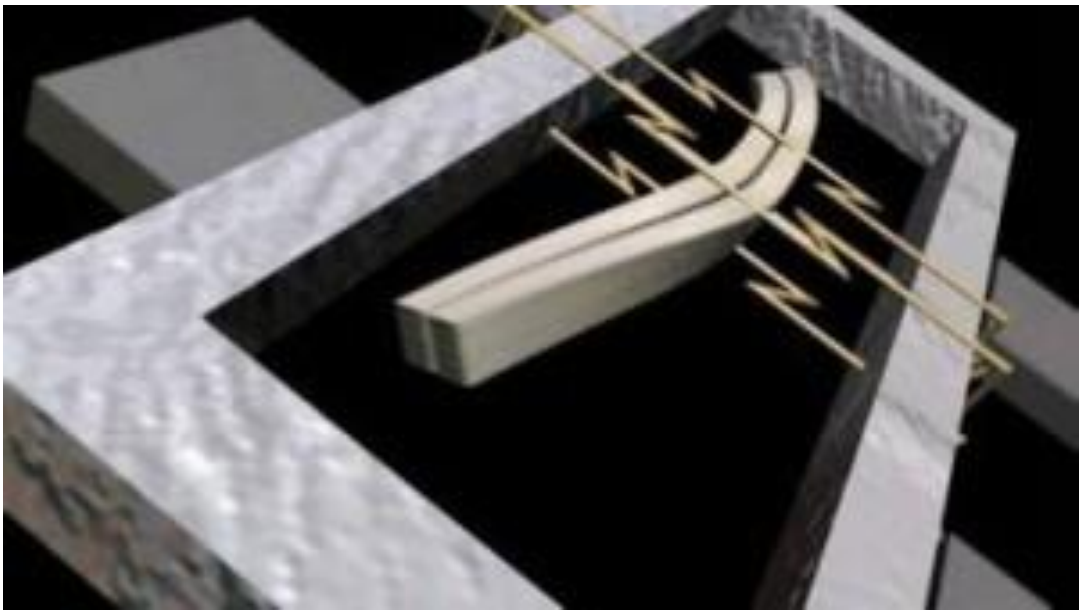
Plusieurs capteurs entrent en jeux: un accéléromètre, un gyroscope, un magnétomètre, un altimètre. Ils travaillent sur 6 degrés de liberté, et fournissent 6 valeurs en sortie.

Comment fonctionne l'accéléromètre?



Un accéléromètre fonctionne avec l'effet piézoélectrique. Ici on peut imaginer un cube avec une boule sur la face inférieure. Les faces de ce cube sont équipées de cristaux piézoélectriques. Lors d'un mouvement, la boule bouge et vient toucher une face. Partant de cela, il est possible de déterminer l'évolution d'un mouvement sous 3 axes grâce aux courants piézoélectriques.

Comment fonctionne le gyroscope?



Les gyroscopes fonctionnent avec l'effet Coriolis. Imaginez une structure en forme de fourchette qui se balance d'avant en arrière. Elle est maintenue en place grâce à un courant piézoélectrique. Quand vous bougez l'ensemble, le cristal piézoélectrique est soumis à une force qui dépend du mouvement et de l'inclinaison.

Connection

Schéma de câblage

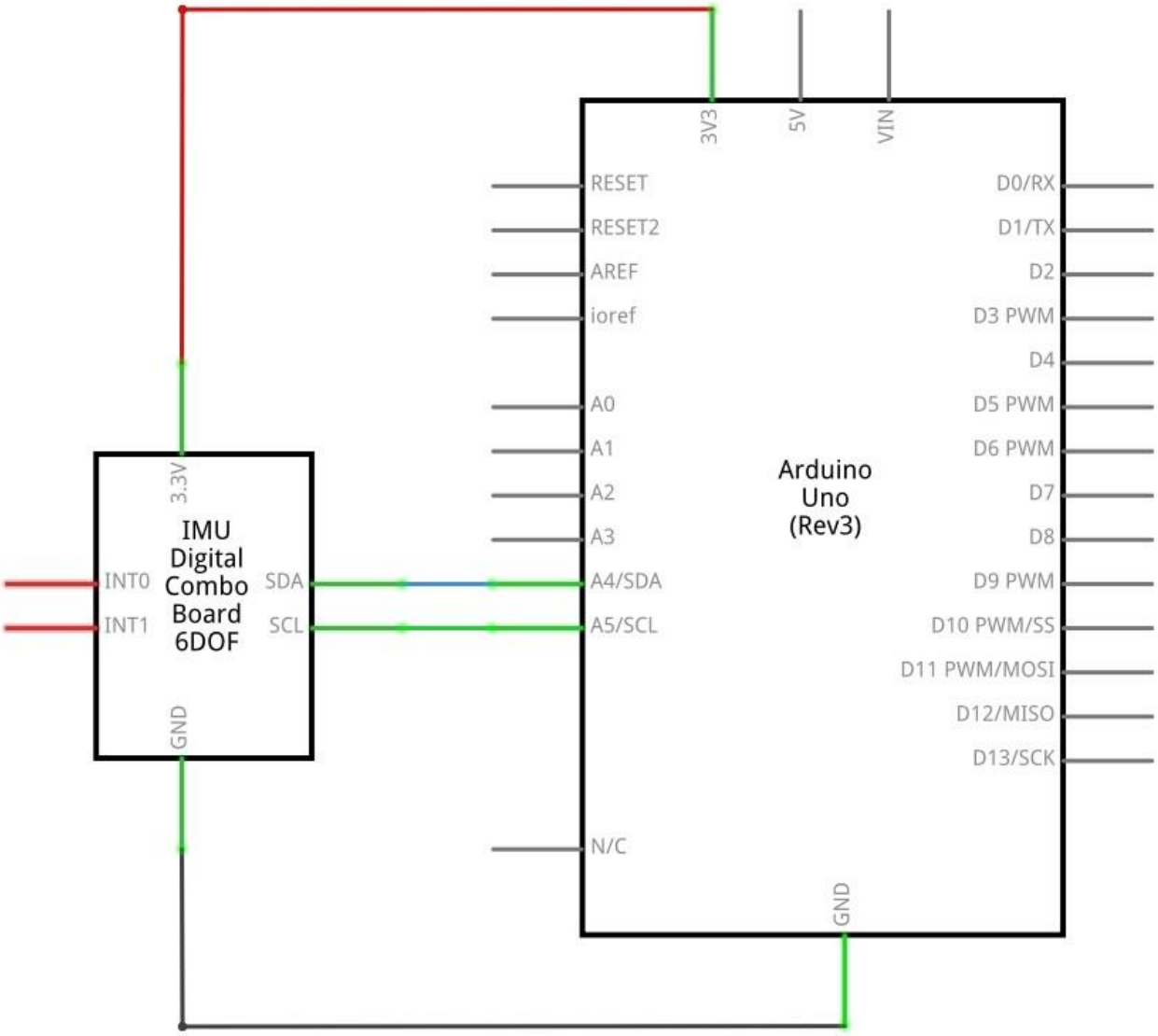
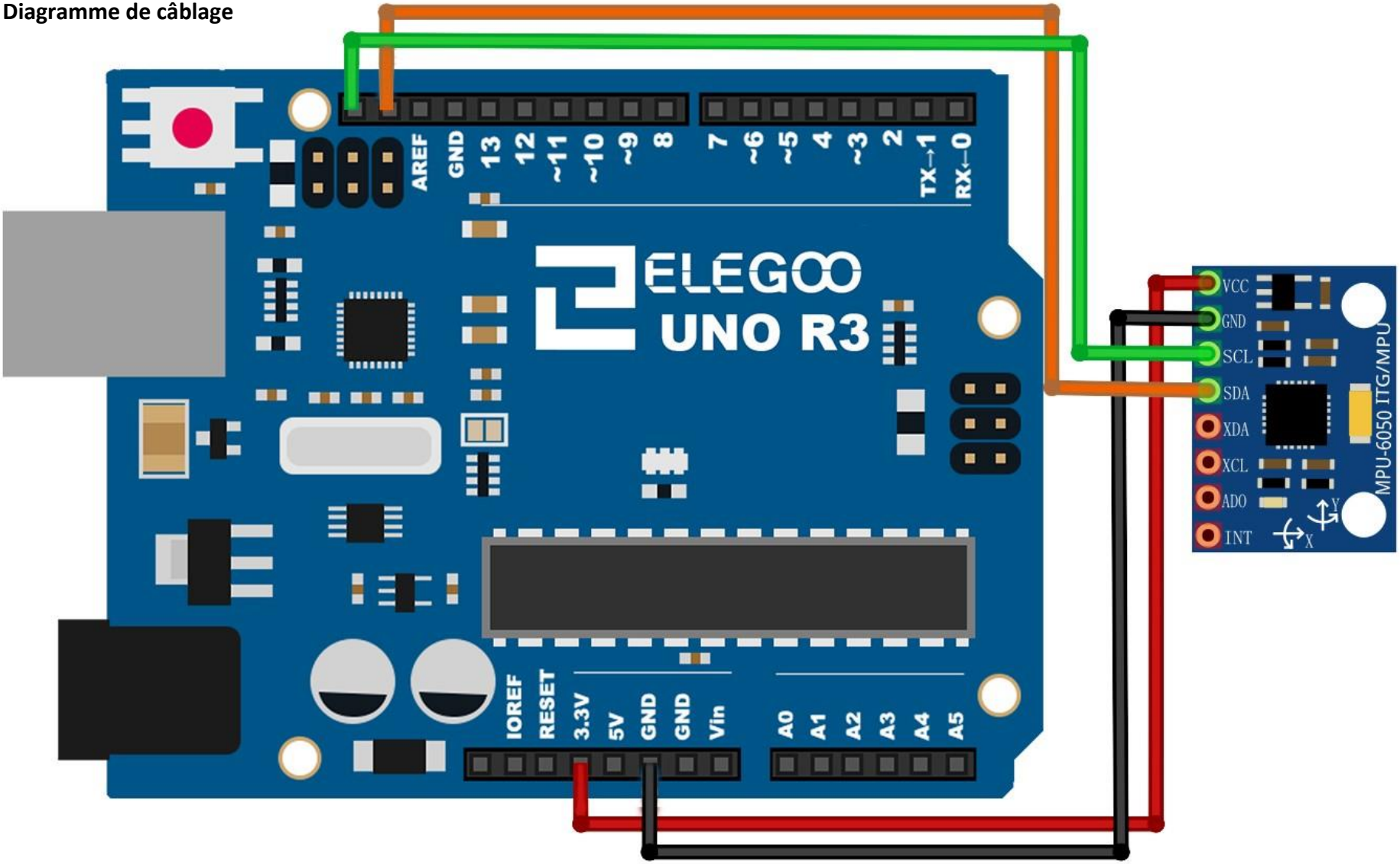


Diagramme de câblage



Ensuite, nous devons configurer les lignes I2C. Pour cela, connectez la broche étiquetée comme SDA sur la GY-521 à la broche analogique 4 d'Arduino (SDA). Et la broche étiquetée comme SCL sur la GY-521 à la broche analogique Arduino 5 (SCL). Et c'est tout, vous avez terminé le câblage de l'Arduino GY-521.

Bibliothèque utilisée

MPU-6050

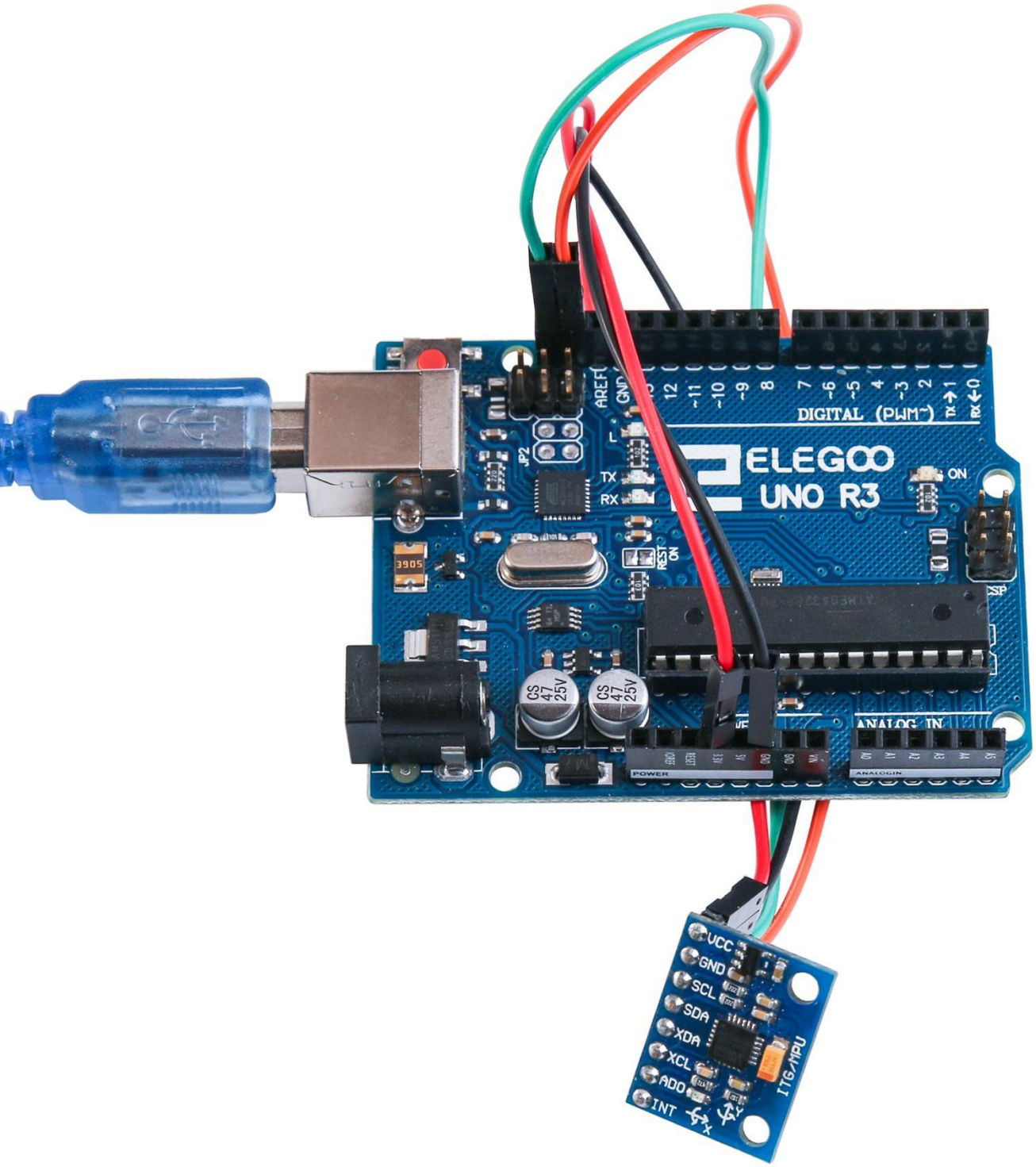
Code

Le code permet de voir les valeurs brutes en sortie de capteur.

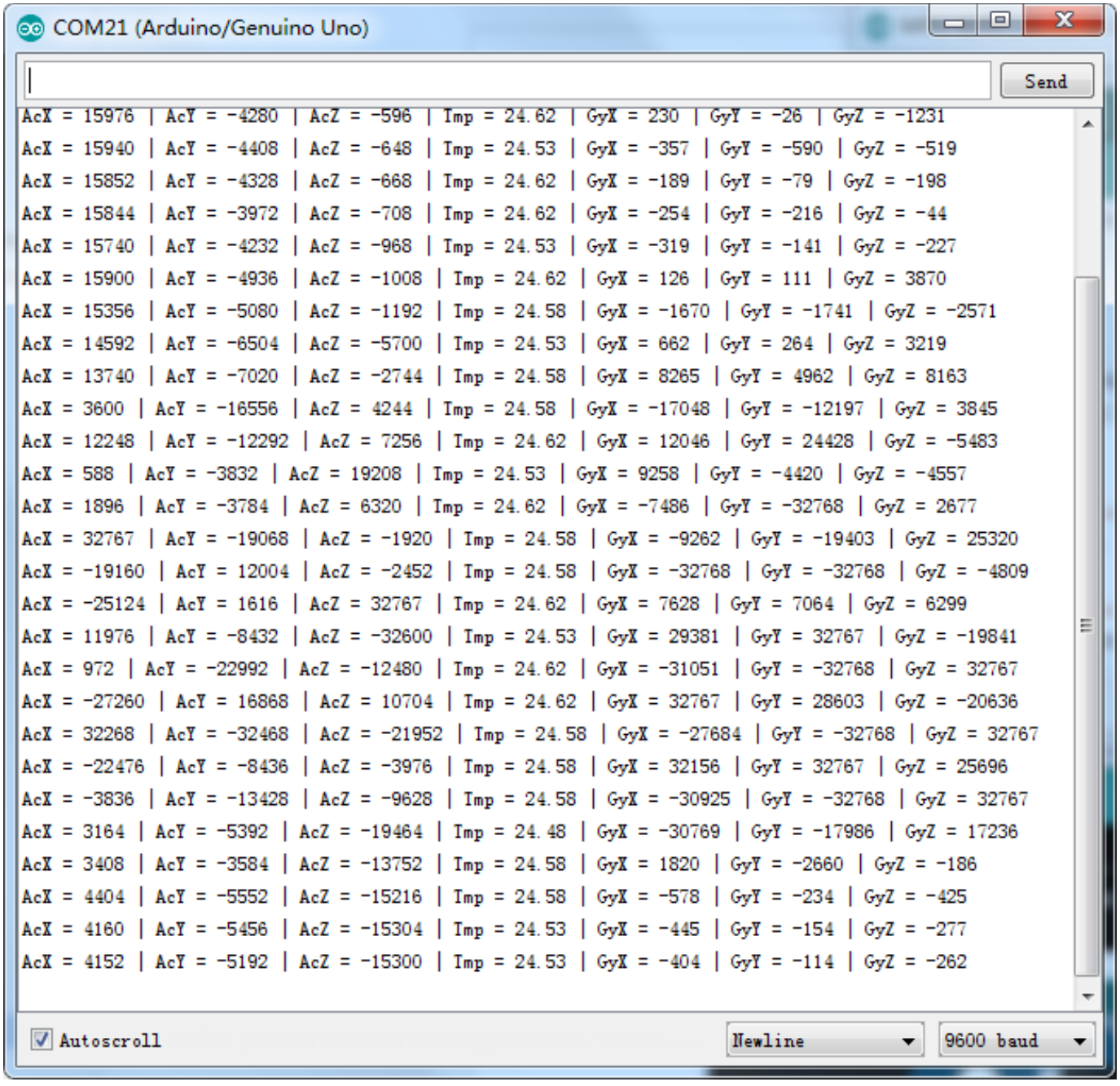
Après avoir réalisé le câblage, ouvrez le sketch “Leçon 16 GY-521 Module » et téléversez-le sur la carte UNO R3.

Veillez bien à avoir installé la bibliothèque < GY-521 > et reportez-vous à la leçon 1 en cas de problème.

Illustration



Ouvrez le moniteur série pour voir les valeurs renvoyées par la carte.



Leçon 17 HC-SR501 PIR Sensor

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de faire de la détection de mouvement avec une carte UNO R3 et le capteur qui va bien. Dans ce projet, une Led sera allumée ou éteinte en fonction de la détection ou non de mouvement.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Détecteur HC-SR501

(3) x Câbles Mâle-Femelle

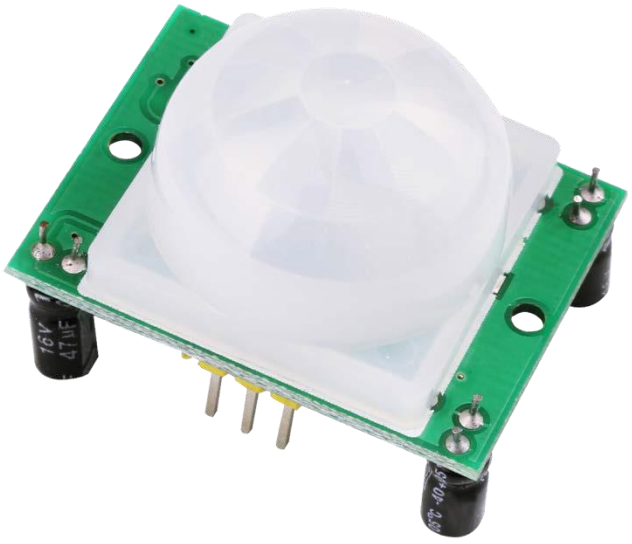
Présentation du composant

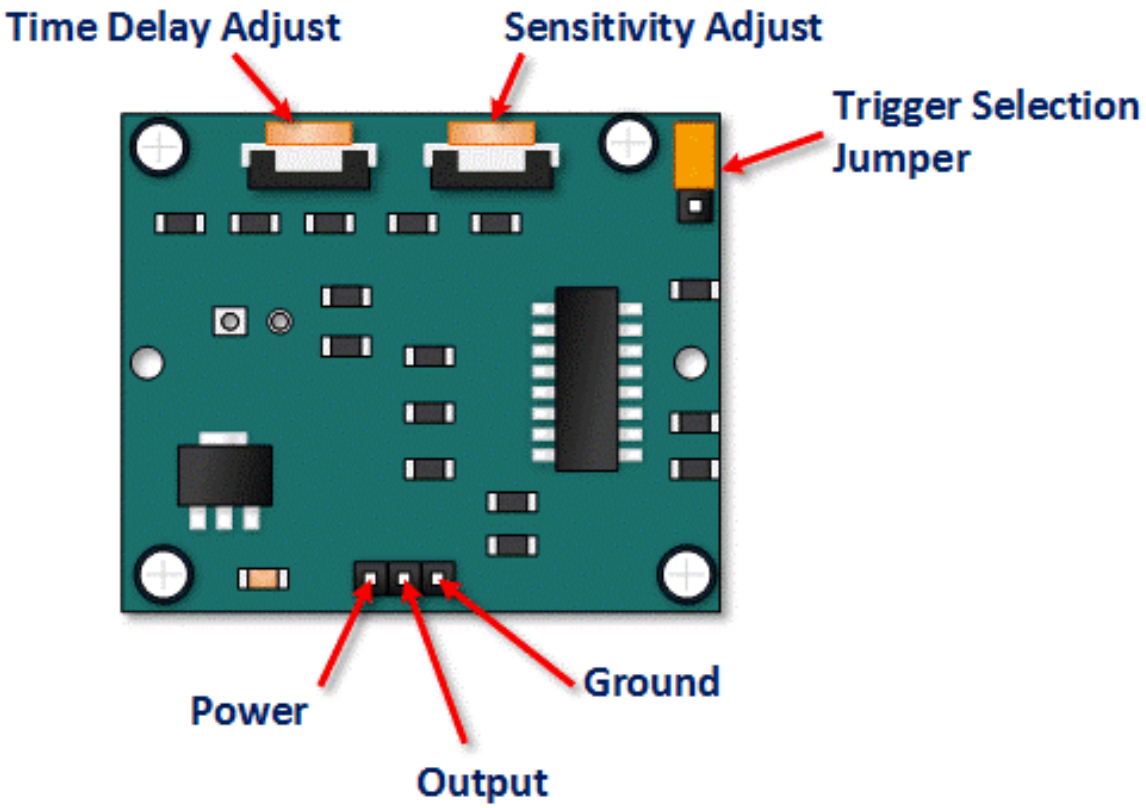
Détecteur PIR:

Il s'agit certainement du capteur le plus compliqué du kit. Un grand nombre de variables entrent en compte pour la gestion de la détection.

Le capteur en lui-même possède 2 slots, chacun étant fait dans un matériau spécial sensible aux infrarouges. La lentille présente ici ne joue pas un grand rôle et le capteur ne peut pas réellement voir beaucoup plus loin que ces capteurs.

Quand le détecteur est en attente, les deux slots reçoivent la même quantité d'infrarouge. Quand « un corps chaud » passe à proximité, cela perturbe l'équilibre des infrarouges : il y a détection.





Pin or Control	Function
Time Delay Adjust	Détermine combine de temps la sortie reste à l'état haut après une détection : de 5s à 5min.
Sensitivity Adjust	Détermine la sensibilité du capteur : jusqu'à 7m
Trigger Selection Jumper	Permet de switcher entre simple et multiple détection(s).
Ground pin	Masse
Output Pin	OUTPUT
Power Pin	+Vcc de 5 à 20V

Fonctionnement du HC SR501 PIR

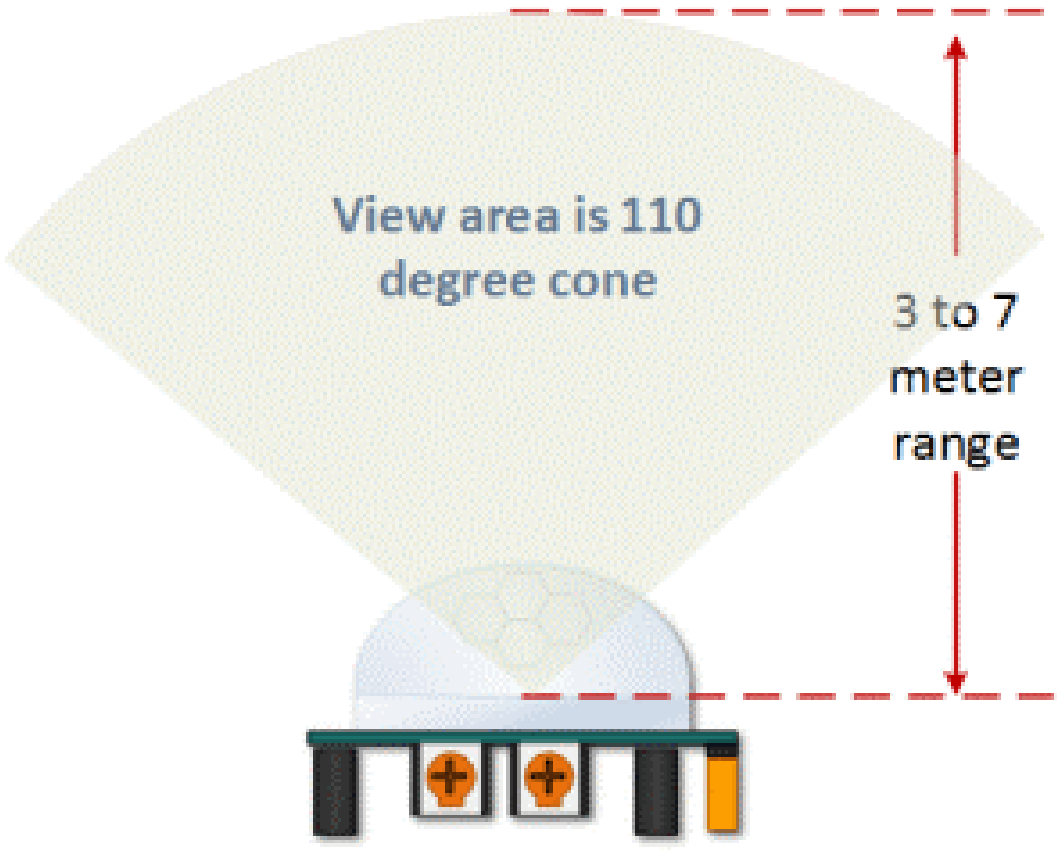
Le capteur va détecter les variations infrarouges de l'environnement. Si il considère qu'il s'agit d'un mouvement, il passe la sortie à l'état haut.

Initialisation

Le capteur a besoin d'une minute pour s'initialiser. En utilisant la partie software, on interdira une détection pendant ce temps.

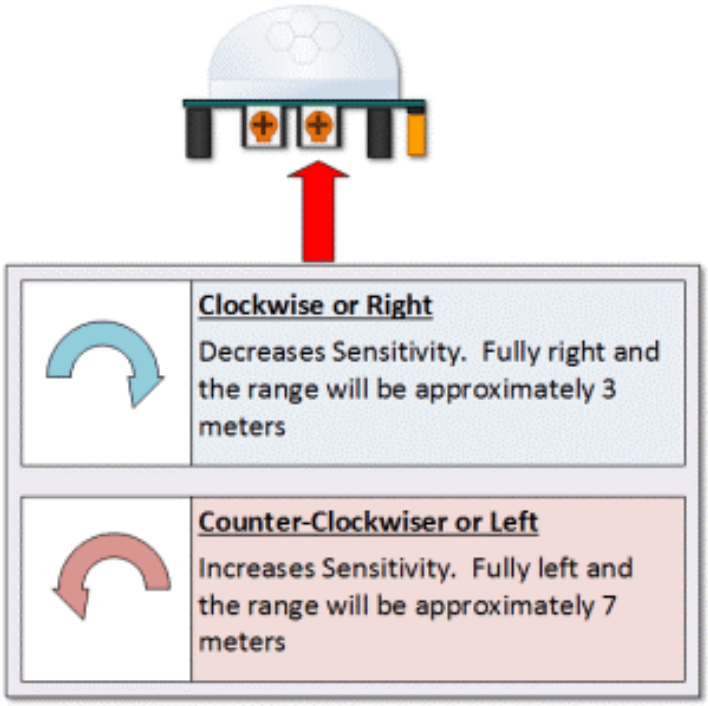
Champ de détection

Le capteur détecte les objets dans le volume illustré ci-après.



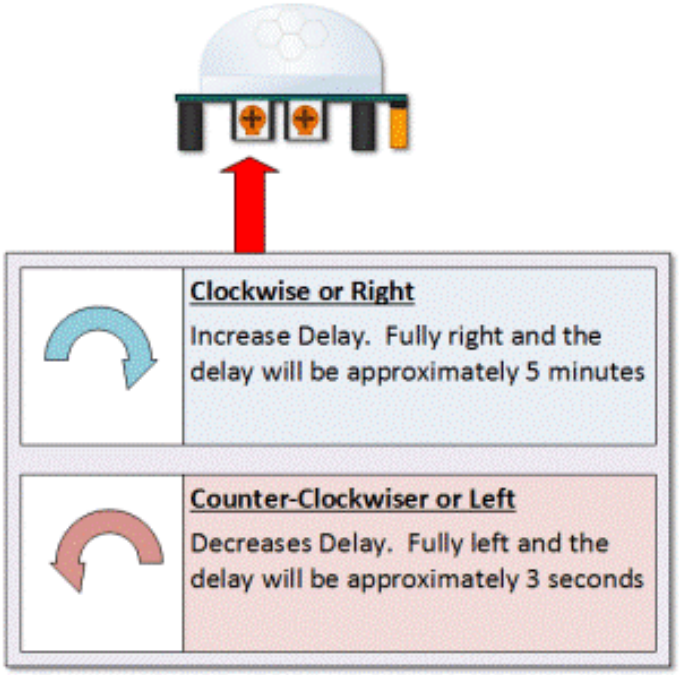
Champ de détection du HC SR501

L'illustration suivante montre comment ajuster la zone de détection:



Délai du HC SR501

L'image suivante montre comment ajuster le temps à l'état haut de OUTPUT après une détection.



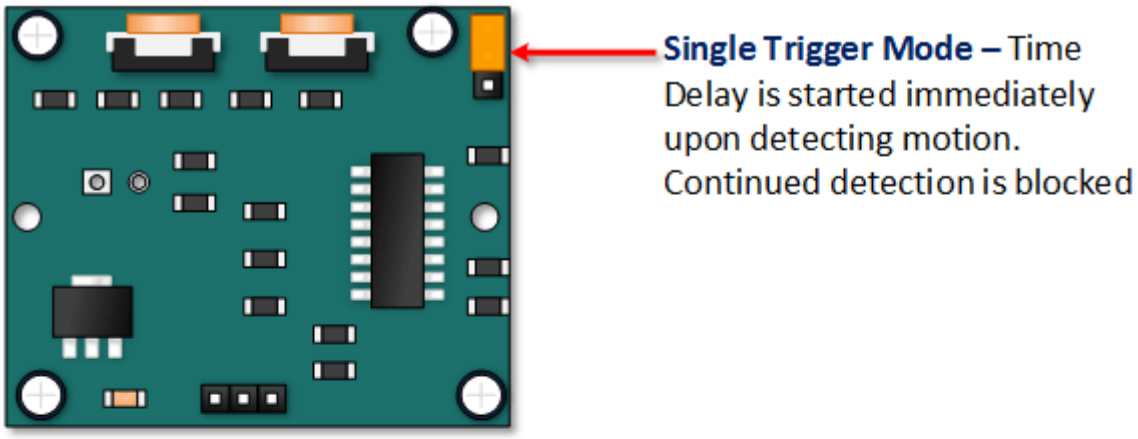
Temps entre deux détections

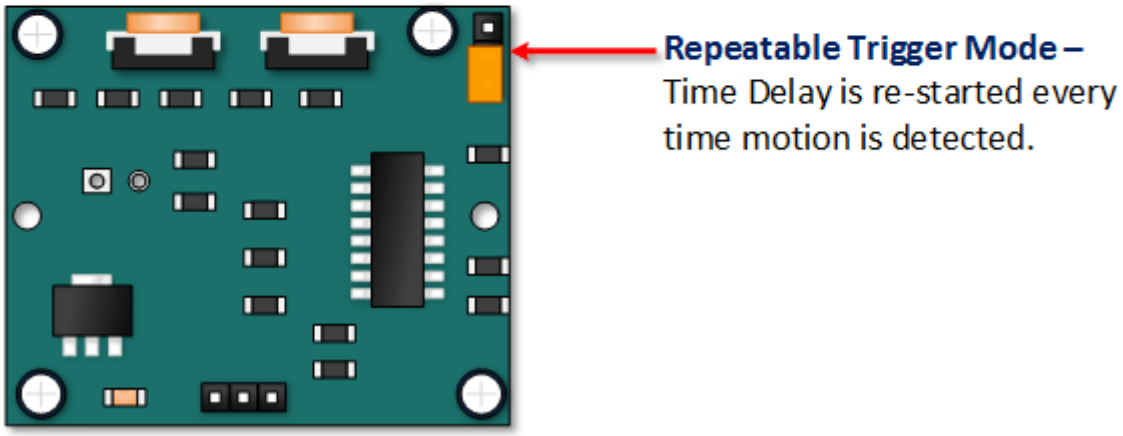
Après la fin d'une détection (lorsque OUTPUT repasse à l'état bas), le capteur sera insensible au mouvement pendant 3S.

Trigger Mode

Grâce à un "Jumper", il est possible de déterminer une détection simple ou des détections répétées.

- **SINGLE TRIGGER** – simple détection.
- **REPEATABLE TRIGGER** – détections multiples.



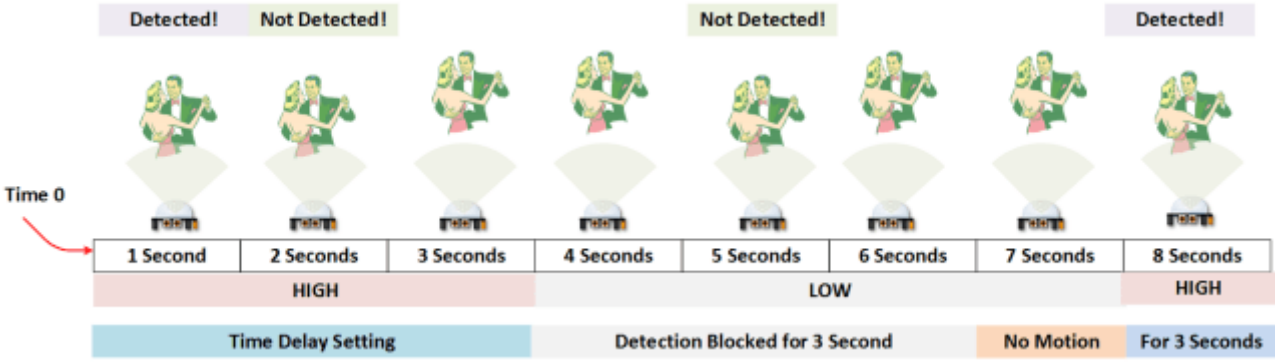


Exemple de la piste de dance

Imaginez que vous voulez contrôler une lumière sur une piste de dance. Comprendre comment les délais de détections et les déclenchements interagissent sera nécessaire pour déterminer comment la lumière va interagir.

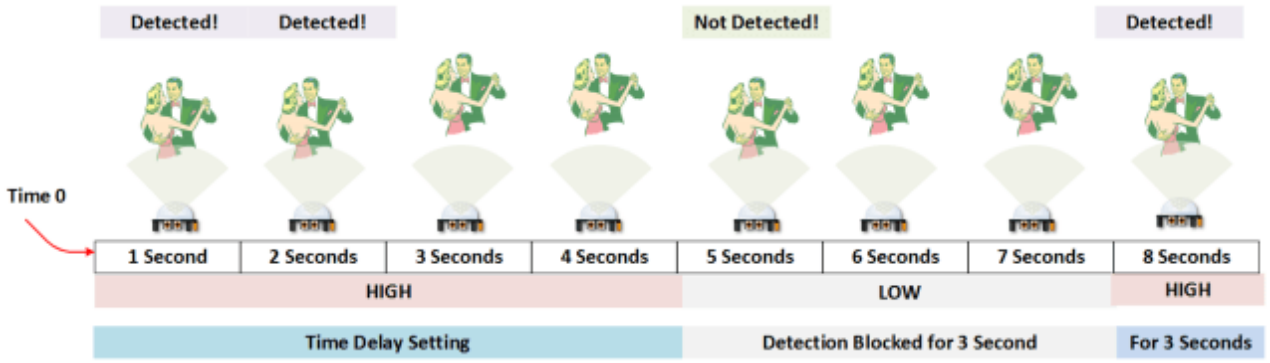
Exemple 1

Dans ce premier exemple, le temps est réglé à 3s et **déclenchement simple**. Comme vous pouvez le voir sur l'illustration, le mouvement initial déclenche l'état haut pendant 3s et qu'il faut attendre le passage à l'état bas pour que la détection soit relancée une nouvelle fois.



Exemple 2

Dans ce nouvel exemple, on change simplement le déclenchement pour le **passer à multiple**. Vous pouvez observer que tant que le mouvement est détecté, l'état reste à haut, il faut que le mouvement ne soit plus détecté pendant 3s avant que la sortie repasse à l'état bas. Cela évite en fait que la lumière s'allume et s'éteigne en permanence.



Connection

Schéma de câblage

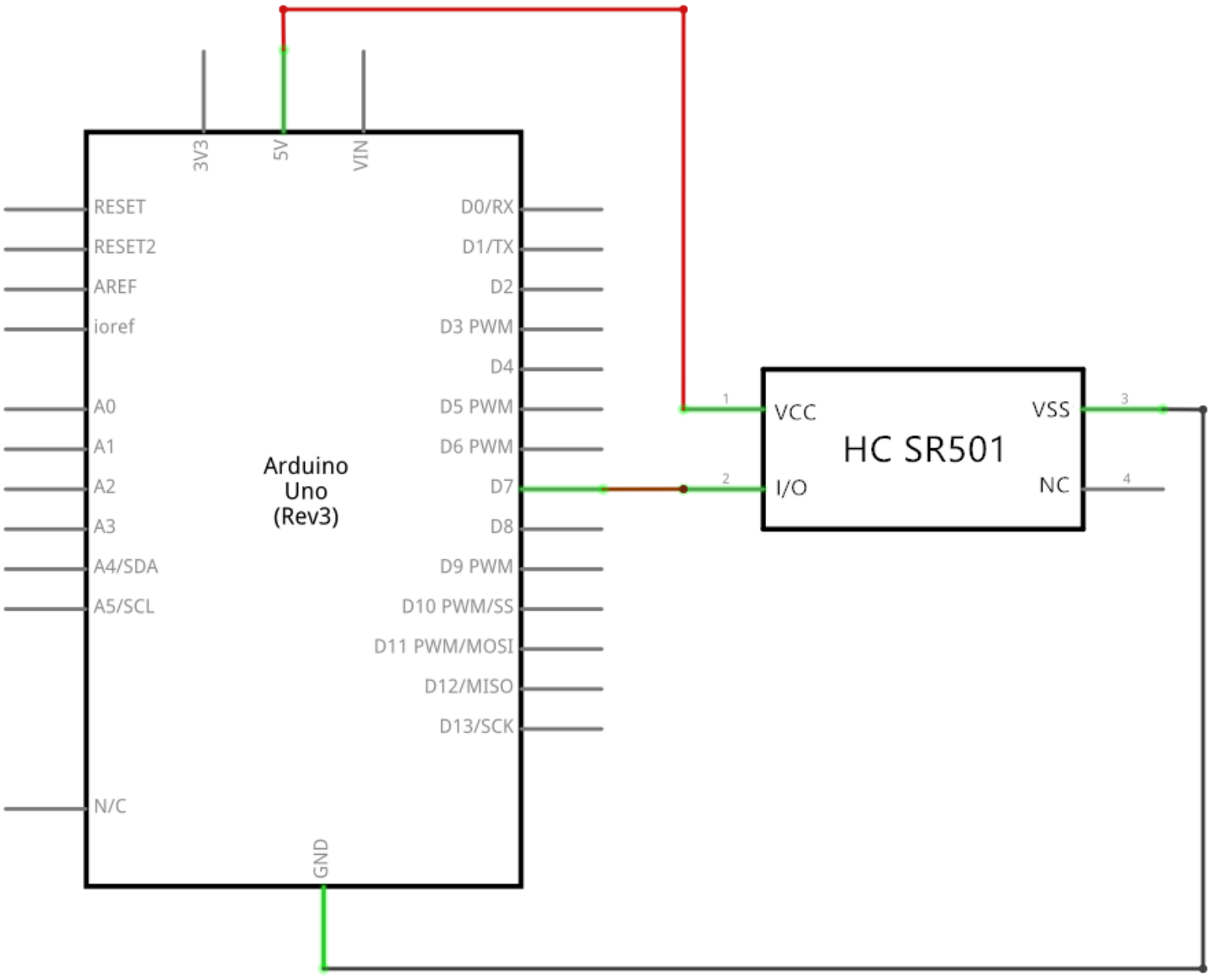
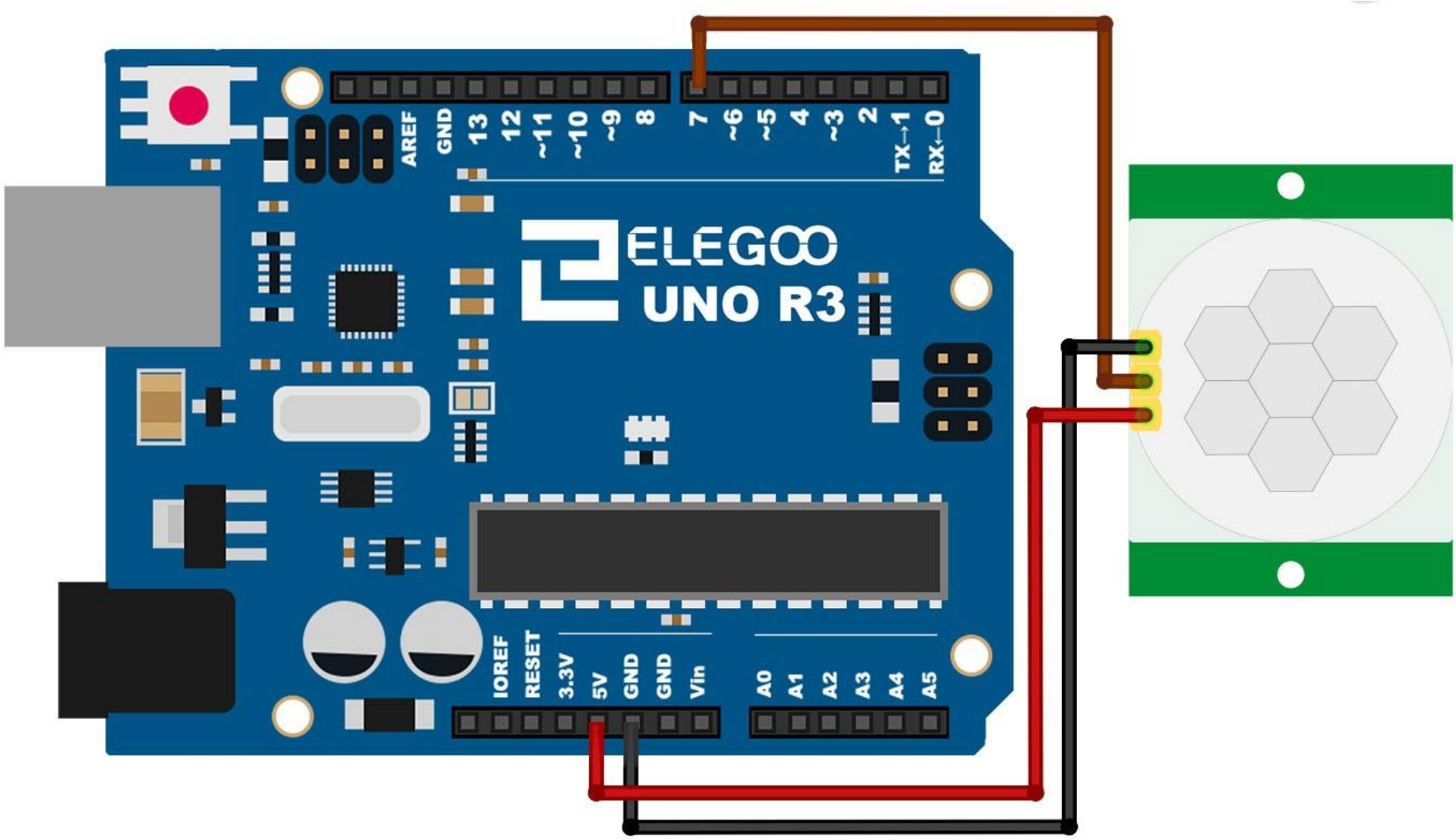


Diagramme de câblage



Comme vous pouvez le constater, le montage est assez simple. Un +5V, une masse et un sortie OUPUT digitale pour avoir un état HAUT ou BAS.

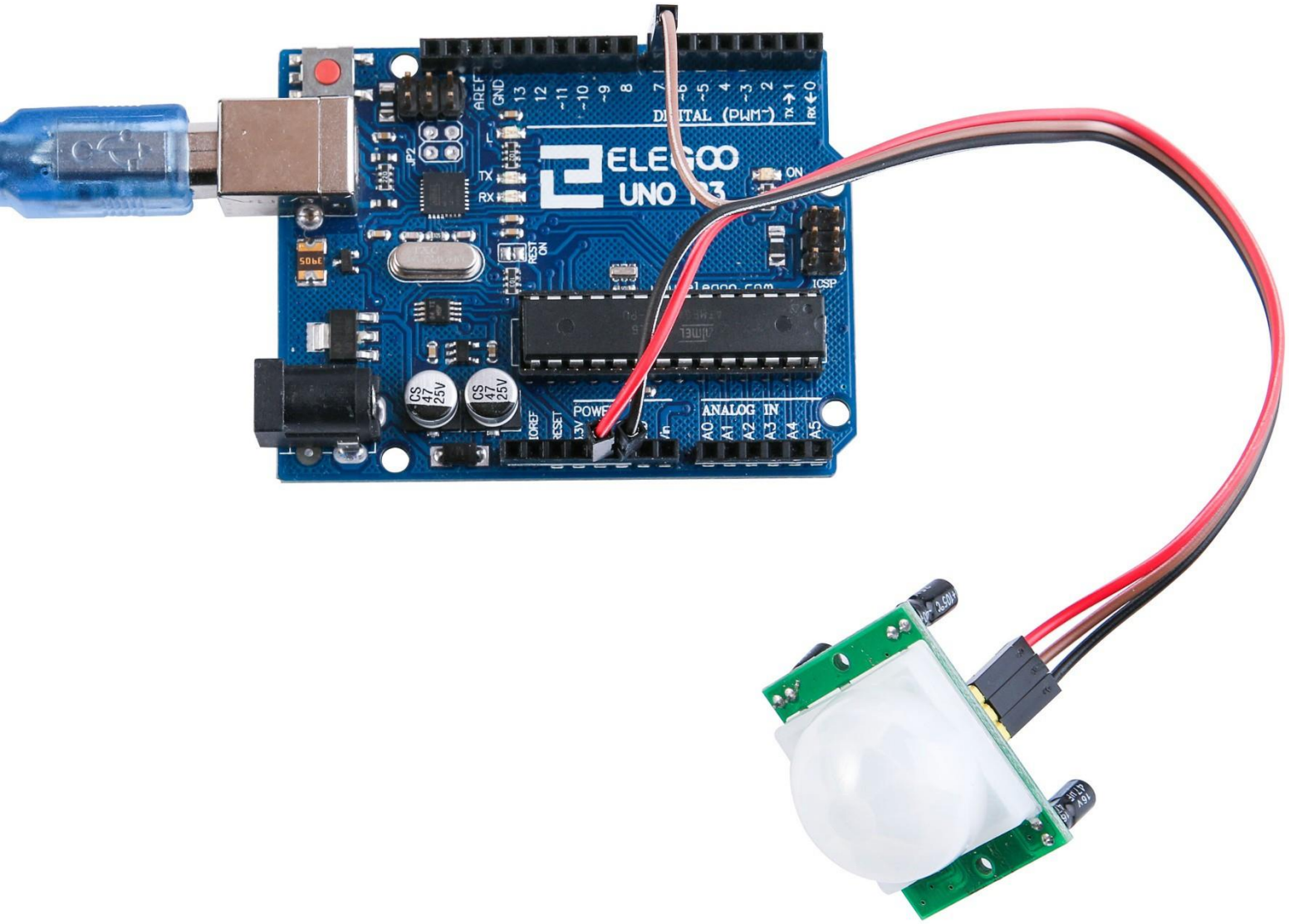
Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 17 HC-SR501 PIR Sensor » et téléversez-le sur la carte UNO R3.

Le sketch allume la LED de la PIN 13 en cas de détection et l’éteint après.

Attention : il faut une minute au capteur pour s’initialiser

Illustration



Leçon 18 Water Level Detection Sensor Module

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser un détecteur de niveau d'eau.

Ce module peut percevoir la profondeur d'eau dans laquelle il est immergé, grâce à une résistance qui va varier.

Matériel nécessaire:

(1) x Elegoo Uno R3

(3) x Câbles Mâle-Femelle

(1) x Water lever detection sensor module

Présentation du composant

Capteur d'eau:

Le module est composé de trois parties. Une partie avec de l'électronique et des pins de connexion, une résistance de pull-up de 1MΩ et des pistes conductrices.

Une série de pistes est reliée à la masse. Entre se trouvent des pistes qui vont permettre de faire la détection et envoyer un signal sur la sortie S.

Les pistes de détection ont une faible résistance de pull-up (1 MΩ). La résistance élimine la valeur de traçage du capteur (signal est à l'état bas) jusqu'à ce qu'une goutte d'eau fasse une mise à la terre et passe la sortie signal à une valeur proportionnelle à la quantité d'eau.

Il a une faible consommation d'énergie et une grande sensibilité.

Caractéristiques: 1、

Voltage: 5V 2、Courant:

<20ma 3、Interface:

Analogique

4、Largeur de détection: 40mm×16mm 5、

Température: 10℃~30℃

6、Signal de sortie: 0~4.2V

Connection

Schéma de câblage

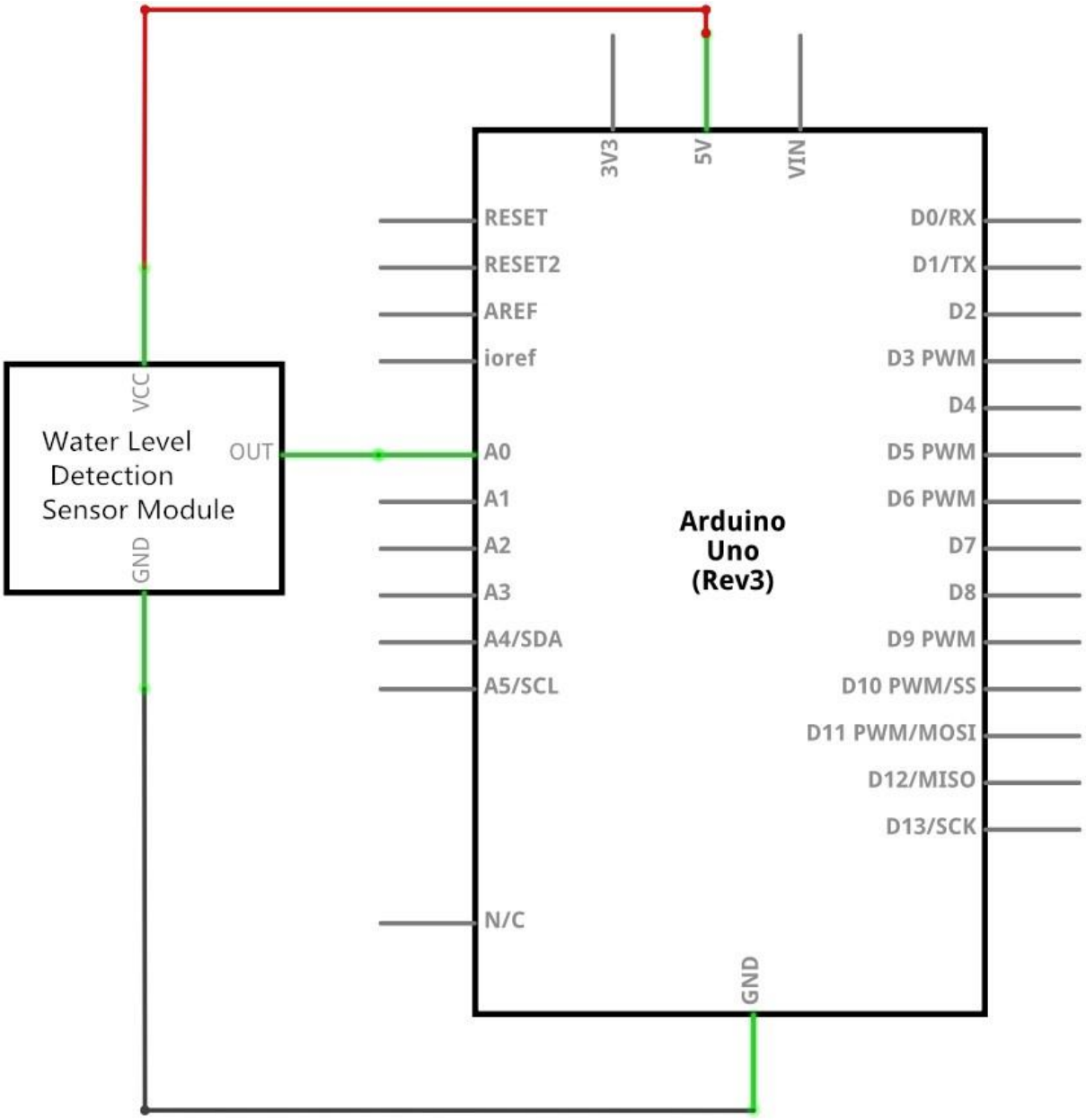
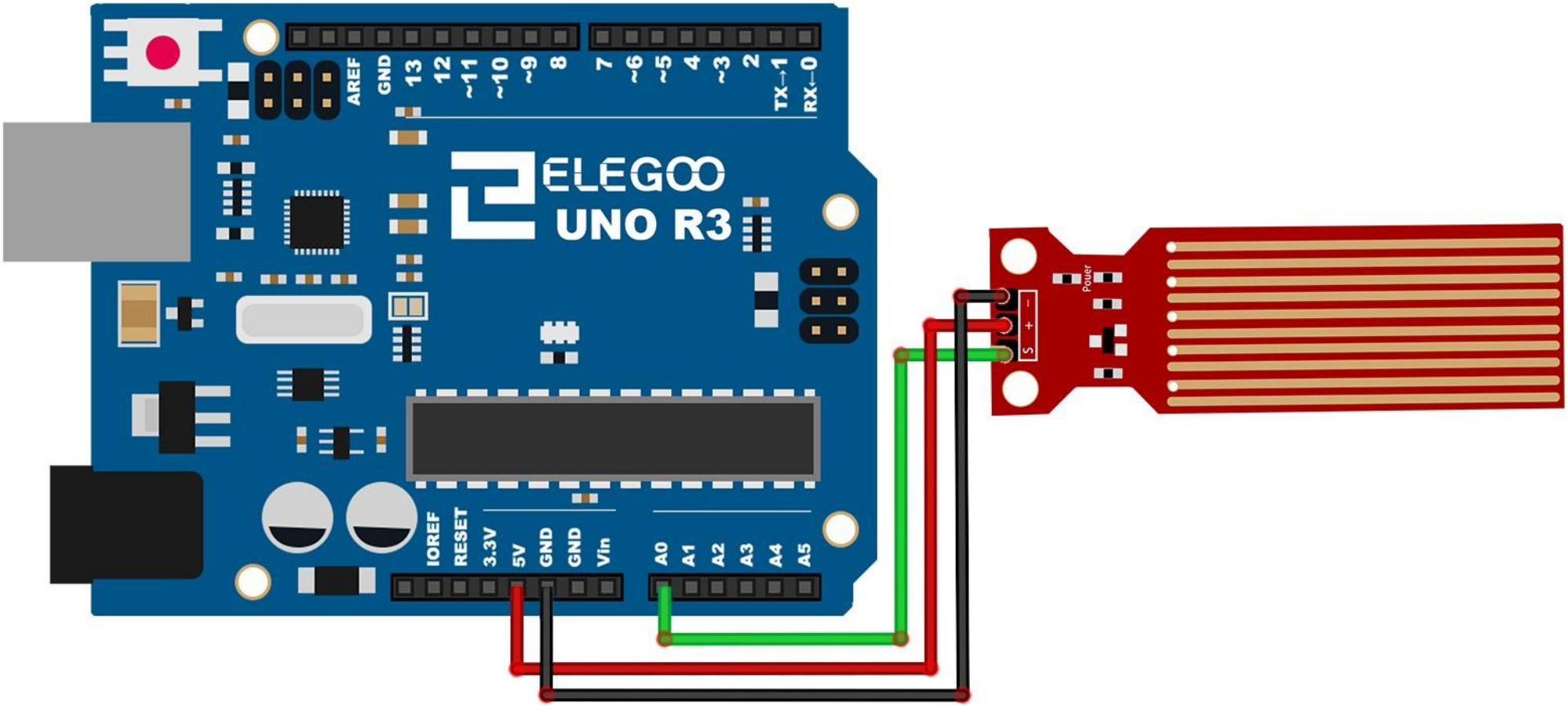


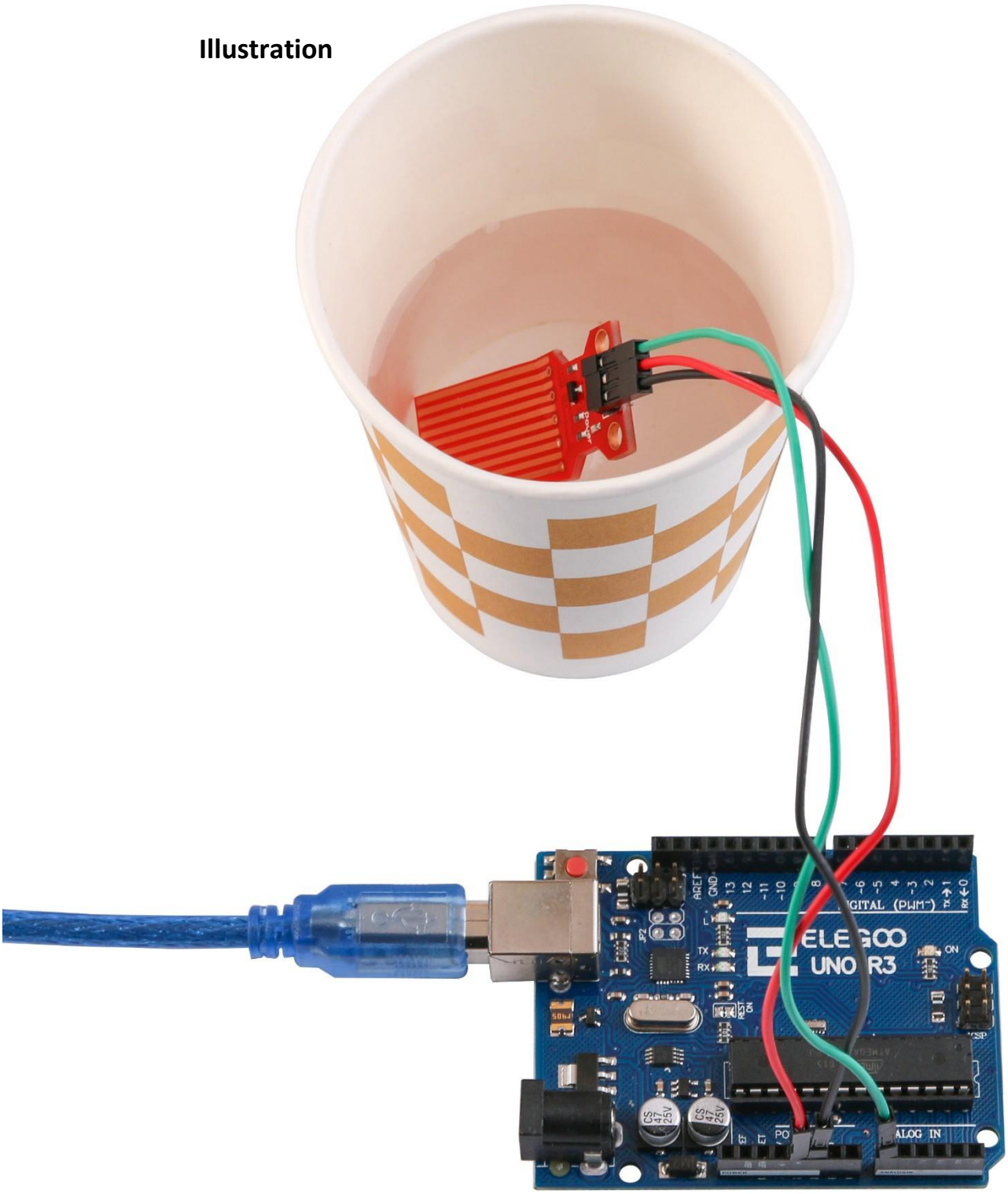
Diagramme de câblage



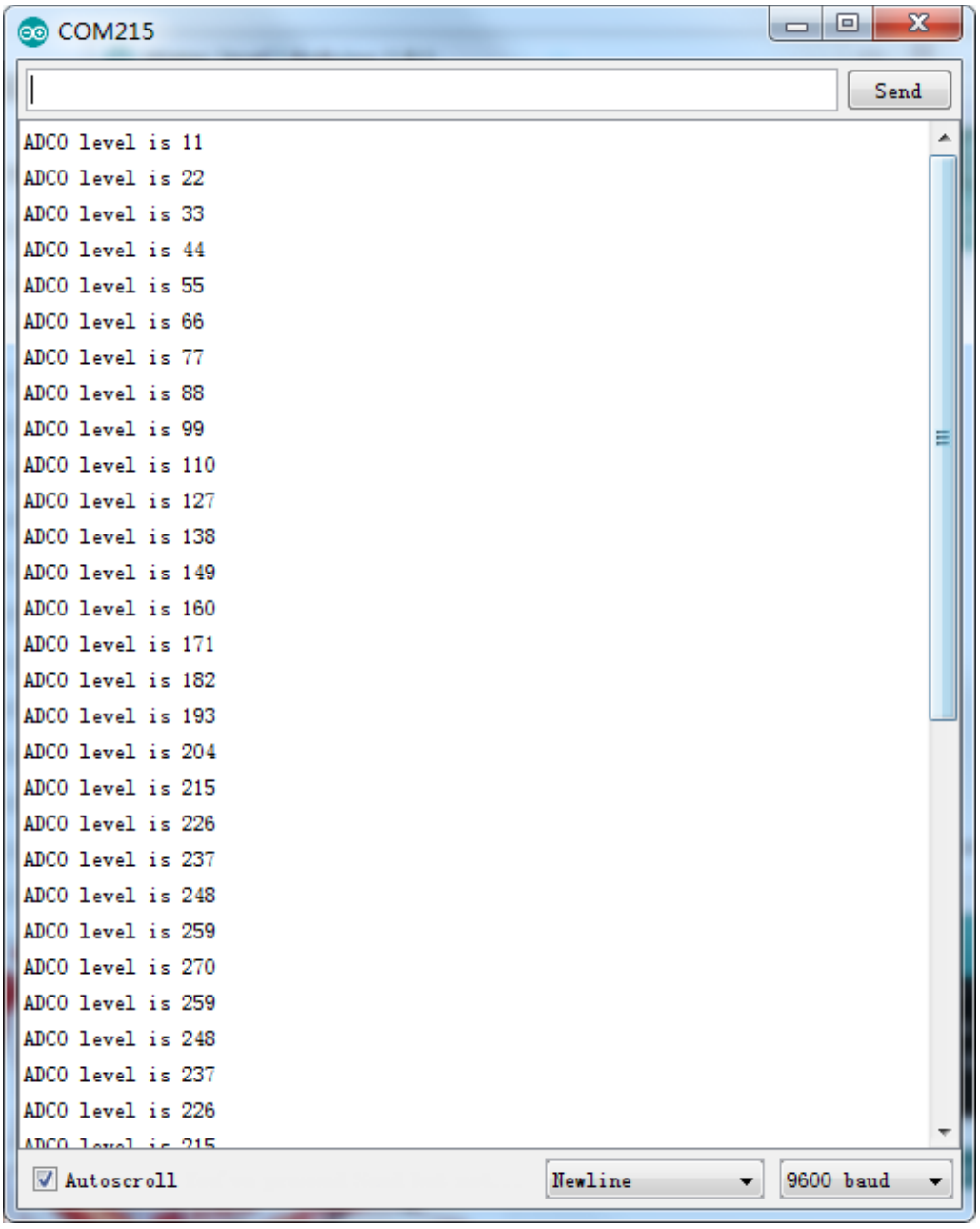
Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 18 Water Level Detection Sensor Module » et téléversez-le sur la carte UNO R3.

Illustration



Ouvrez le moniteur série et observez les valeurs renvoyées par la carte.



Leçon 19 Real Time Clock Module

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser le module DS3231, qui permet de connaitre l'année, le mois, le jour, l'heure, la minute, la seconde et la semaine. Le module est équipé d'une pile pour continuer de fonctionner même en cas de rupture d'alimentation.

Matériel nécessaire:

(1) x Elegoo Uno R3

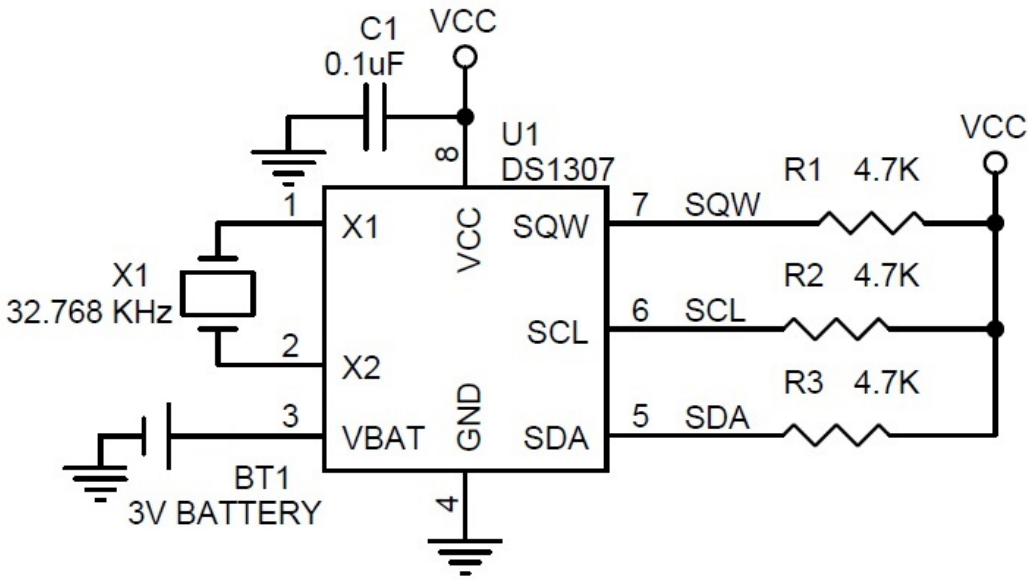
(1) x Module DS3231 RTC

(4) x Câbles Mâle-Femelle

Présentation du composant

DS3231

Il s'agit d'une simple puce de mesure du temps équipée d'une pile pour fonctionner san



NOTICE: If there is the left version of module in the Kit, don't worry, its function and pin names are the same as the new version. Just follow the wiring diagrams and sketch in the tutorial below to get it working.

Connection

Schéma de câblage

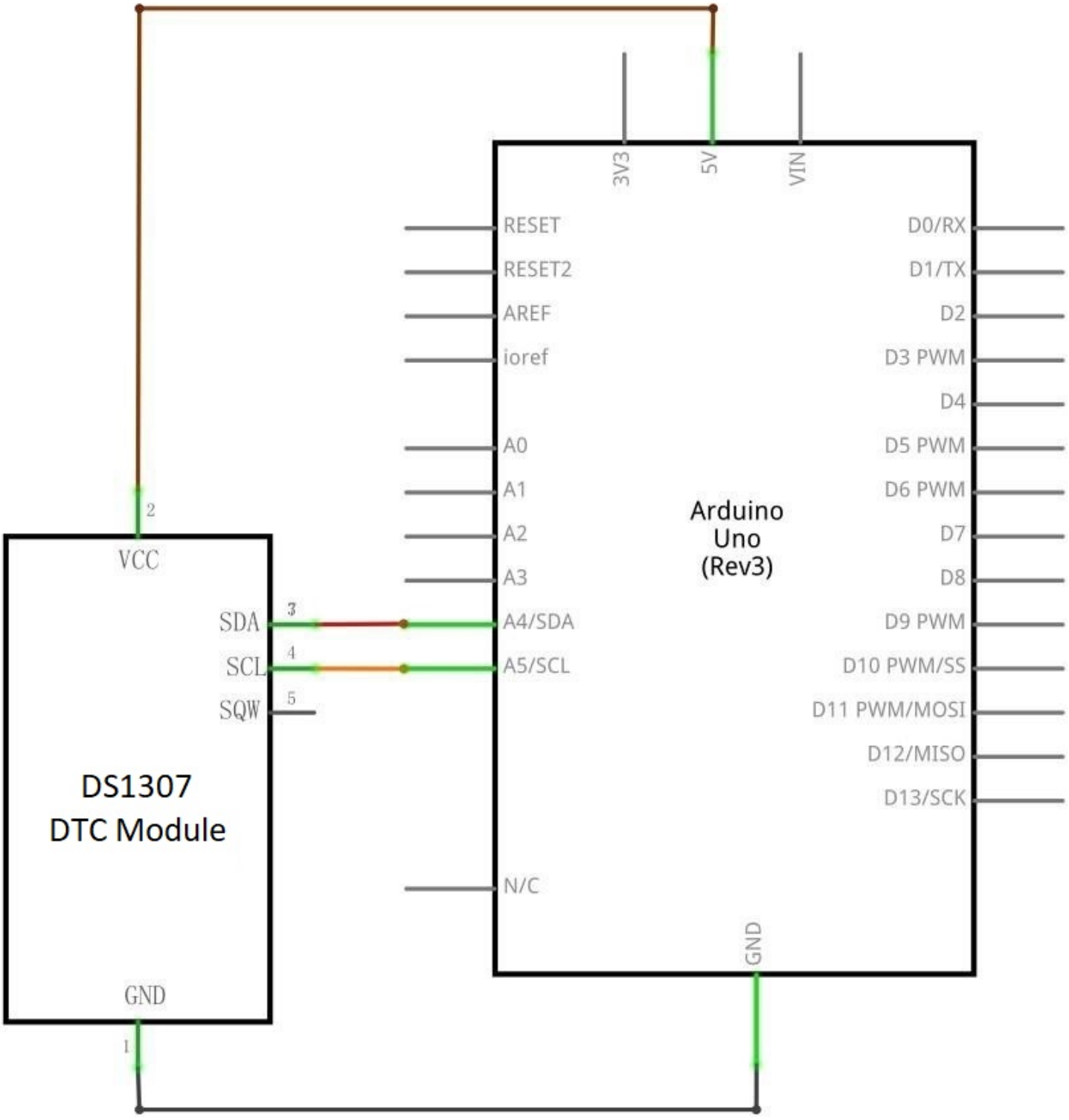
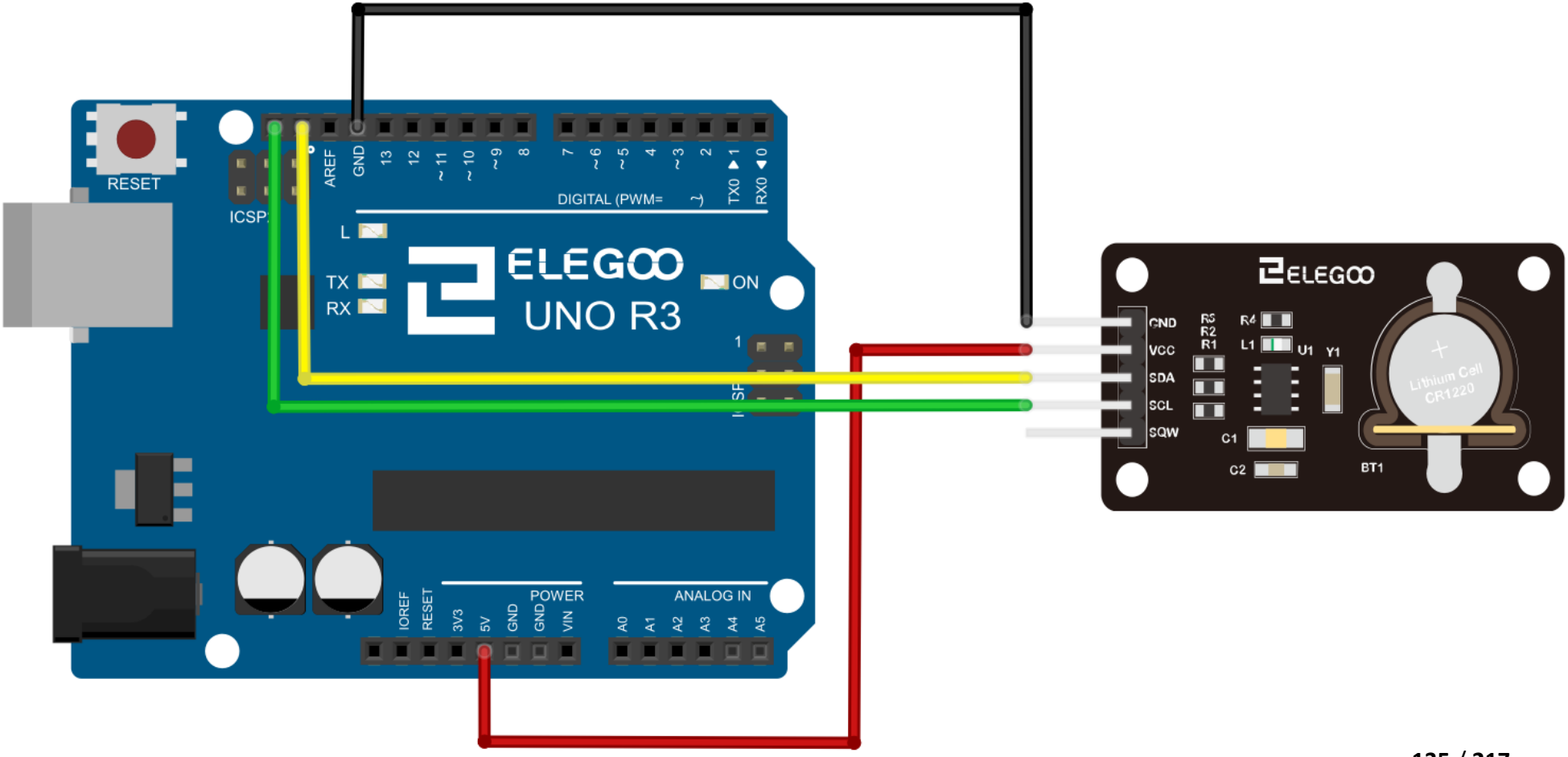


Diagramme de câblage



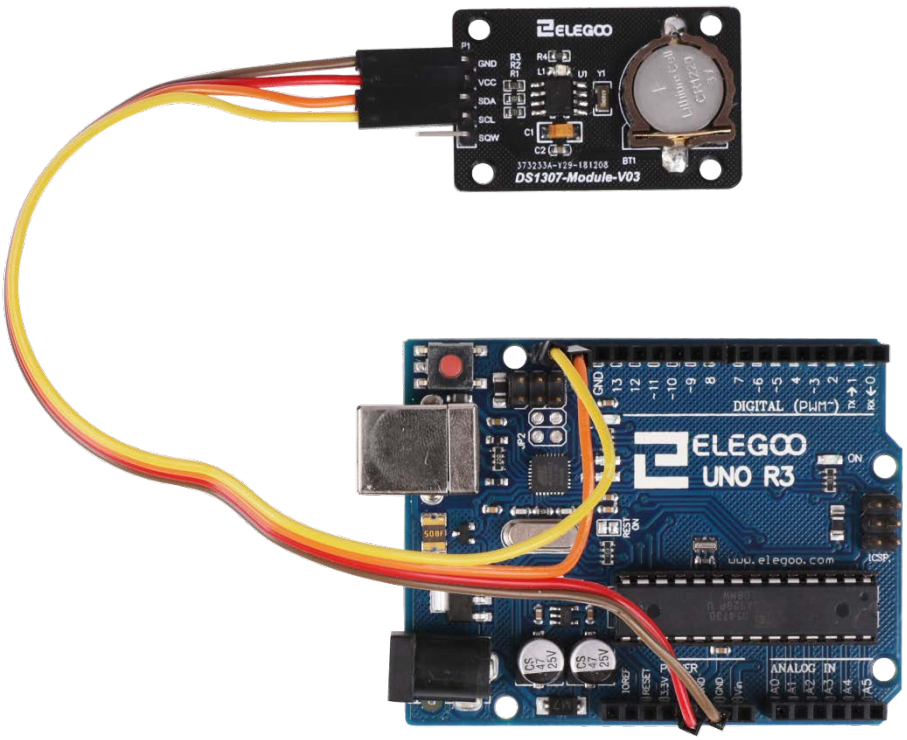
Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 19 Real Time Clock Module” et Téléversez-le sur la carte UNO R3.

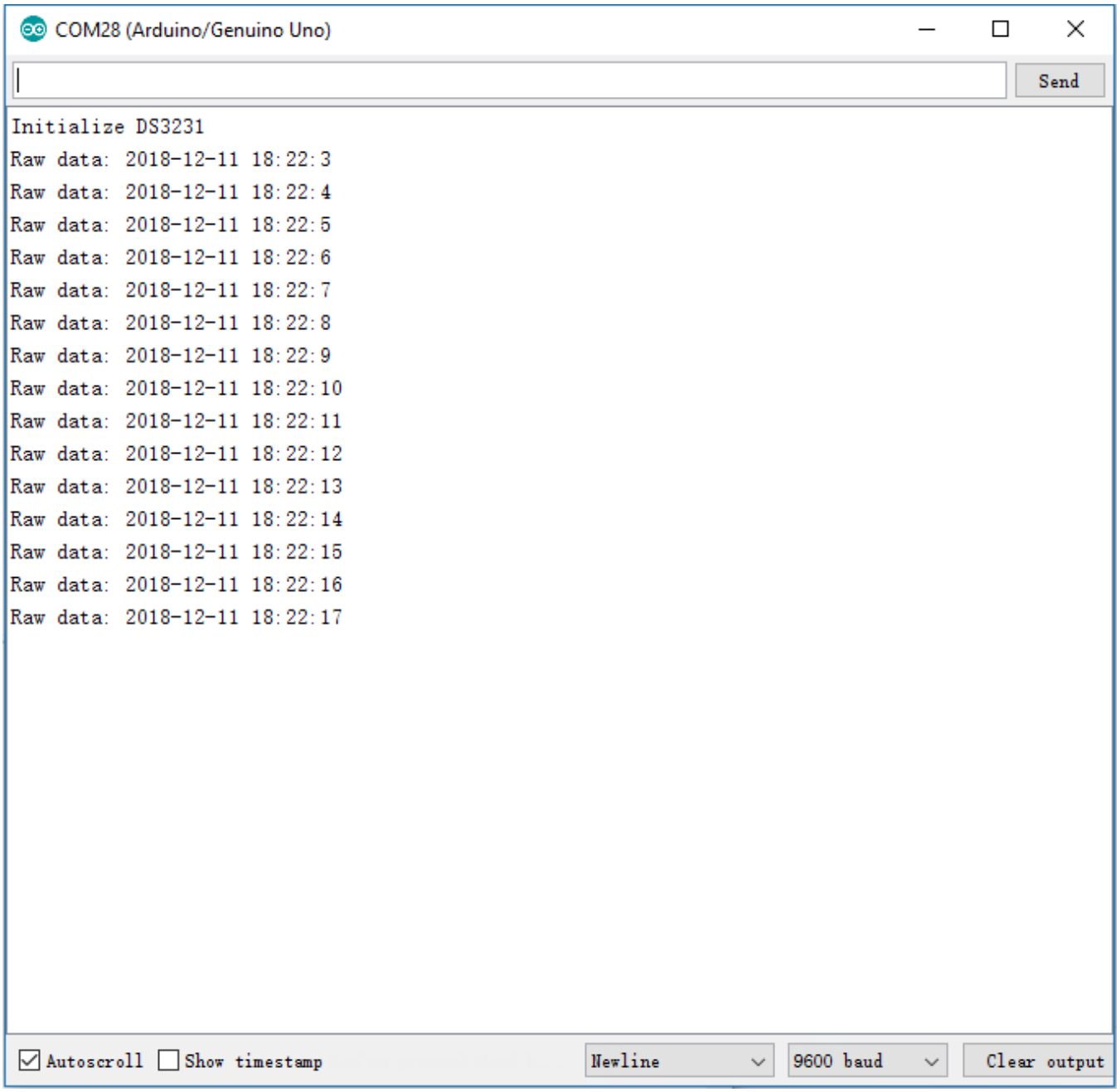
Avant toute chose, vous devez installer la bibliothèque < DS3231 >, référez-vous à la leçon 1 en cas de problème sur l’installation.

.

Illustration



Ouvrez le moniteur série afin d’observer les valeurs renvoyées par la carte.



Leçon 20 Sound Sensor Module

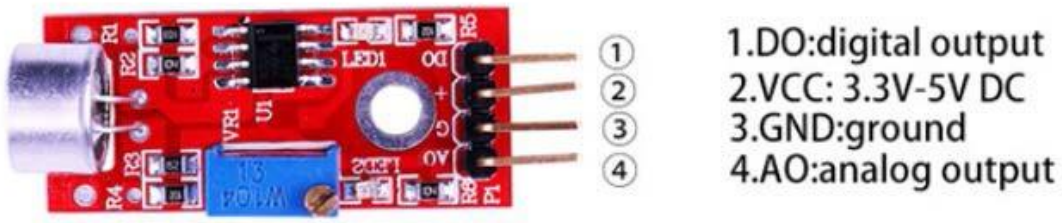
But de la leçon

Dans cette leçon vous allez apprendre à utiliser un capteur de son. Ce module dispose de deux sorties.

AO: sortie analogique. Voltage temps-réel, signal du microphone.

DO: quand l'intensité du son atteint un seuil, la sortie passe à l'état haut. La sensibilité est ajustable via un potentiomètre.

Le potentiomètre est la pièce bleue que vous pouvez voir sur la photo ci-dessous. La petite vis au sommet permet de faire le réglage de seuil.



Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Capteur de son

(4) x Câbles Mâle-Femelle

Présentation du composant

Microphone

Les transducteurs sont des appareils qui permettent de transformer l'énergie d'une forme à une autre. Le microphone permet de transformer une voix en signal électrique. Le fonctionnement est l'inverse du fonctionnement d'un haut-parleur. On en trouve de toutes tailles suivant l'application. Ici nous allons voir le fonctionnement d'un microphone à condensateur électrique qui est largement utilisé dans les téléphones et ordinateurs.

Comme son nom le suggère, le microphone à condensateur est composé de plaques

parallèles et fonctionne sur le principe de la capacitance. L'une des plaques est fixe, l'autre mobile. Le déplacement de l'air généré par l'énergie sonore va se transformer en déplacement de la plaque mobile. Il en résulte un potentiel électrique.

Ces microphones sont largement utilisés dans les applications qui nécessitent la détection de petits sons.

Les composants sont encapsulés dans un corps métalliques La face supérieure est fermées pour protéger de la poussière le diaphragme.

Connection

Schéma de câblage

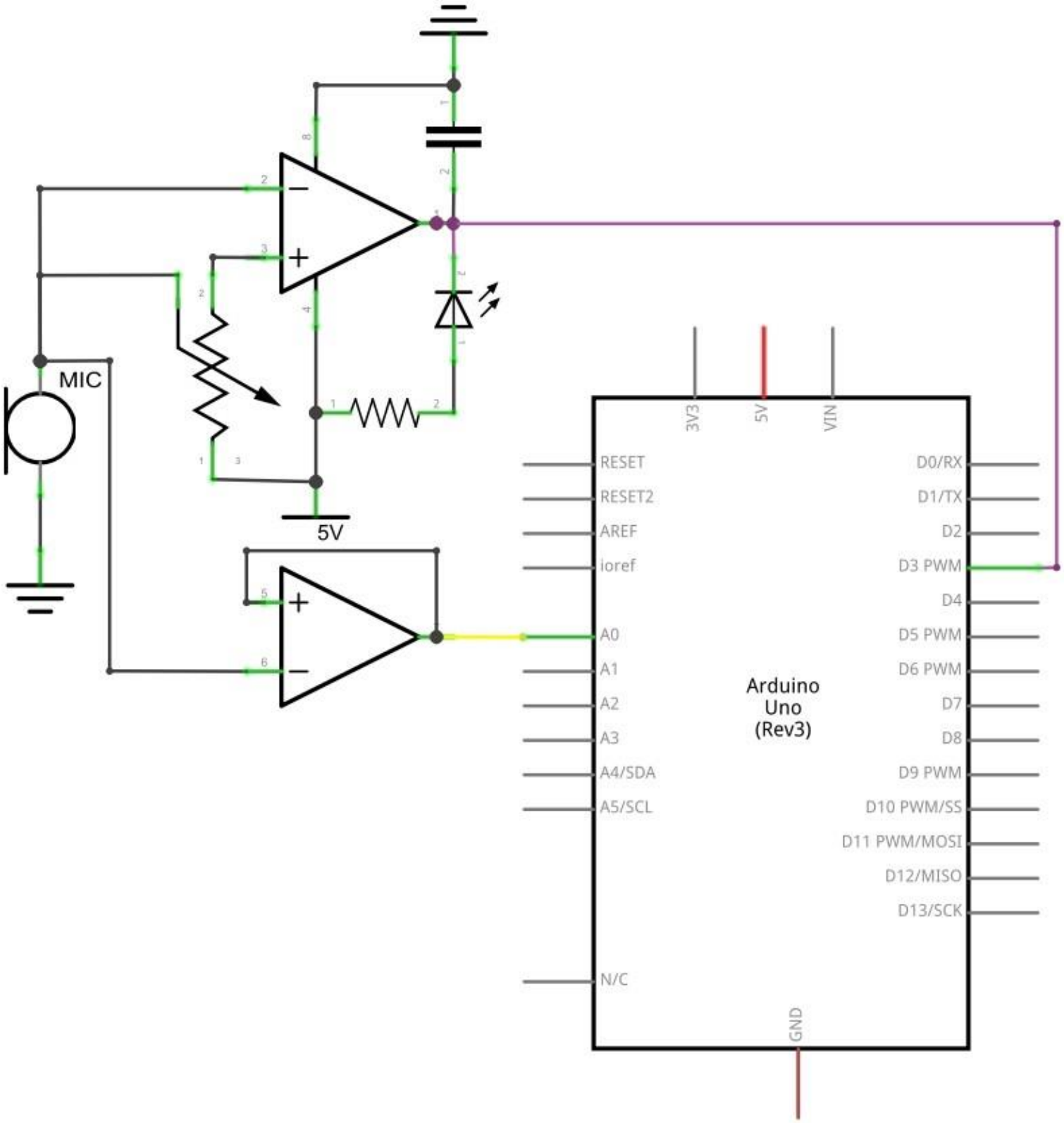
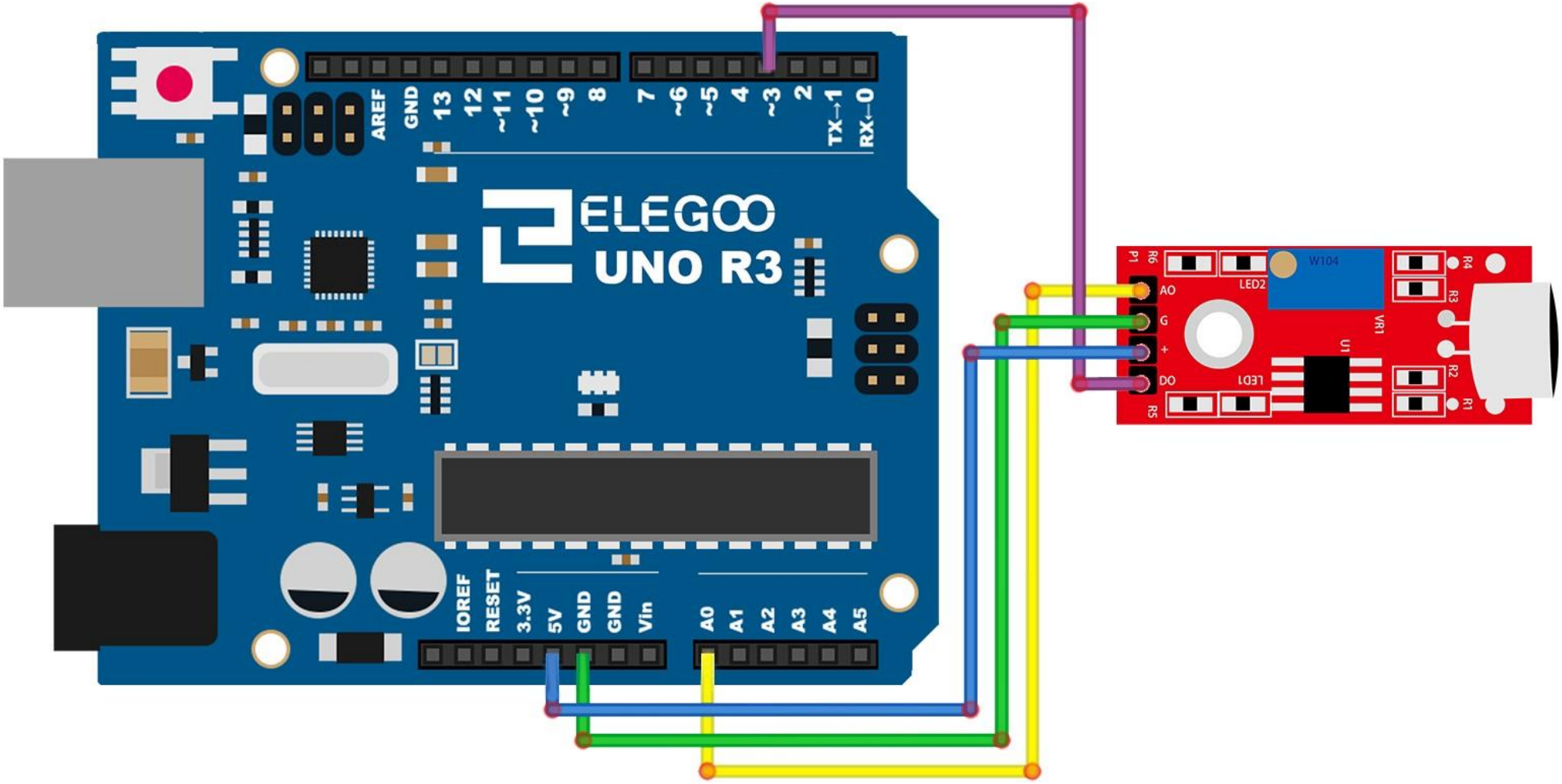


Diagramme de câblage

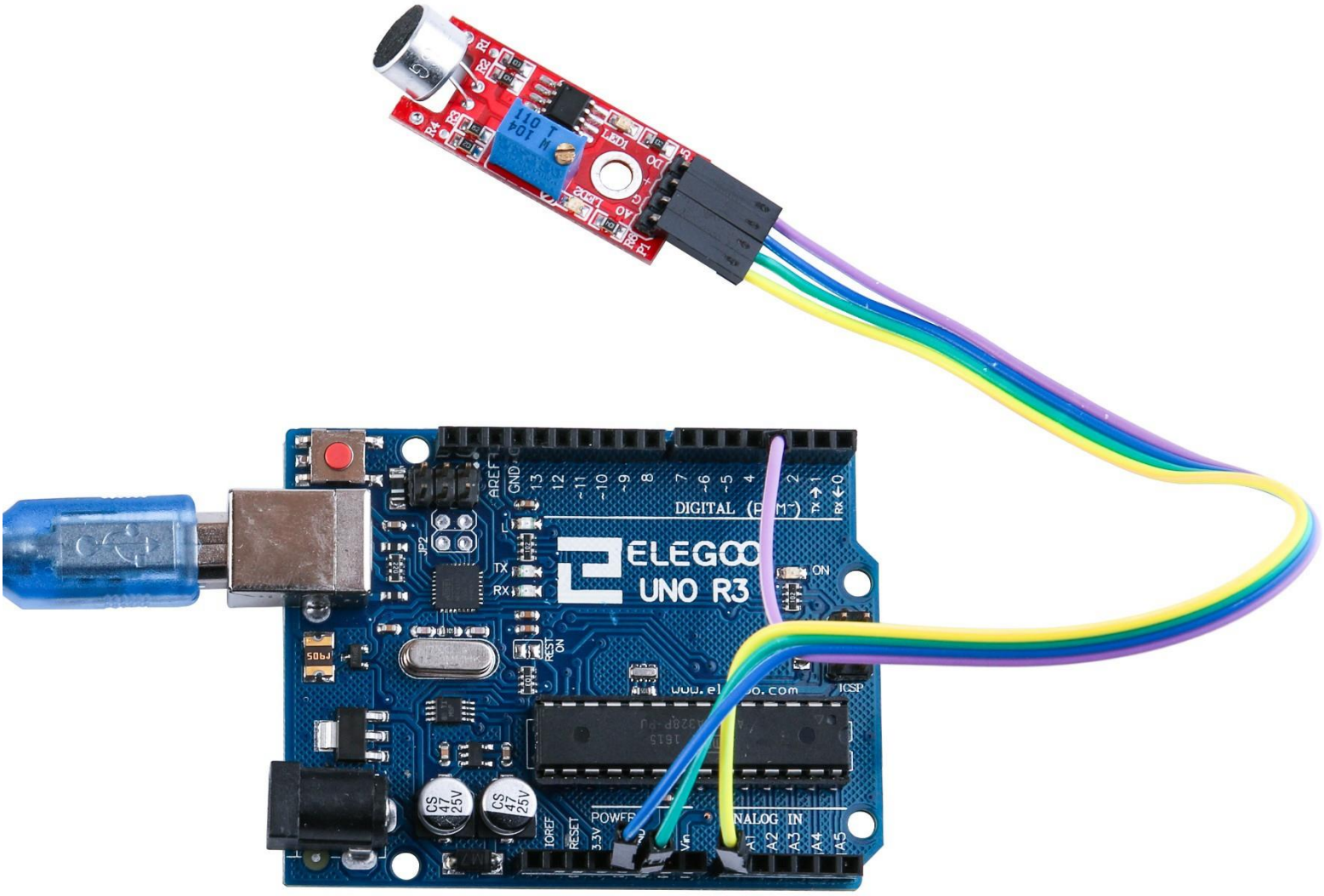


Code

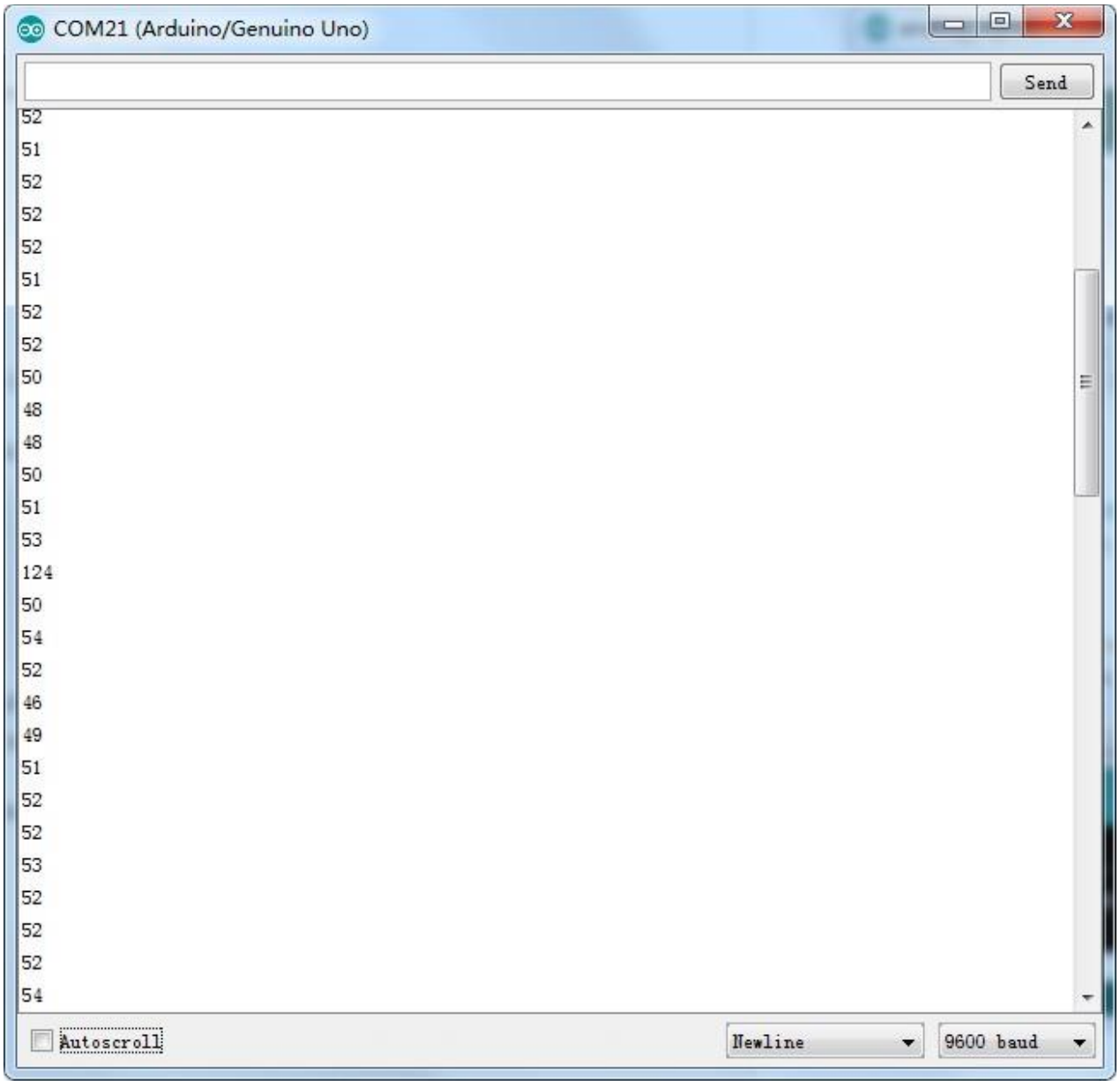
Après avoir réalisé le câblage ouvrez le sketch " Leçon 20 Sound Sensor Module" et Téléversez-le sur la carte UNO R3.

Ce module fourni 2 type de signaux en sortie : digital_signal_output et analog_signal_output. Le code digital_signal_output travail avec un seuil. Lorsque le son atteint un certain niveau, la sortie de la carte passe à l'état haut. Il est possible en jouant avec le potentiomètre de changer la valeur de seuil. Le code analog_signal_output va moduler la valeur de la sortie en temps réel en fonction du son mesuré.

Illustration



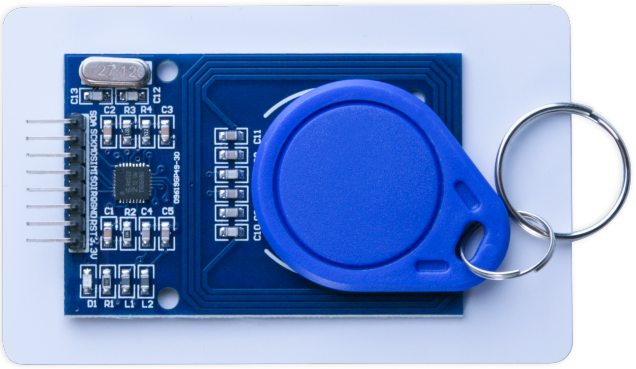
Ouvrez le moniteur série et observez les valeurs renvoyées par la carte.



Leçon 21 RC522 RFID Module

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser un lecteur de carte RFID.



Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Module RC522 RFID

(7) x Câbles Mâle-Femelle

Présentation du composant

RC522

Le MFRC522 est un module permettant de faire de lecture et de l'écriture en RFID qui travaille à 13.56 MHz. Il travaille sur avec la norme ISO 14443A / MIFARE®.

Le MFRC522 ne nécessite pas de système de réception/émission supplémentaire. Le récepteur fourni une implémentation robuste et efficace de démodulation et décodage du signal selon la norme ISO/IEC 14443A/MIFARE. Le MFRC522 supporte les communications sans contact utilisant le système de transfert haut débit MIFARE® à 848 kbit/s en flux montant et descendant.

Plusieurs interfaces hôtes sont intégrées:

• Interface SPI

• Interface UART (similaire au RS232)

• Interface I2C.

Connection

Schéma de câblage

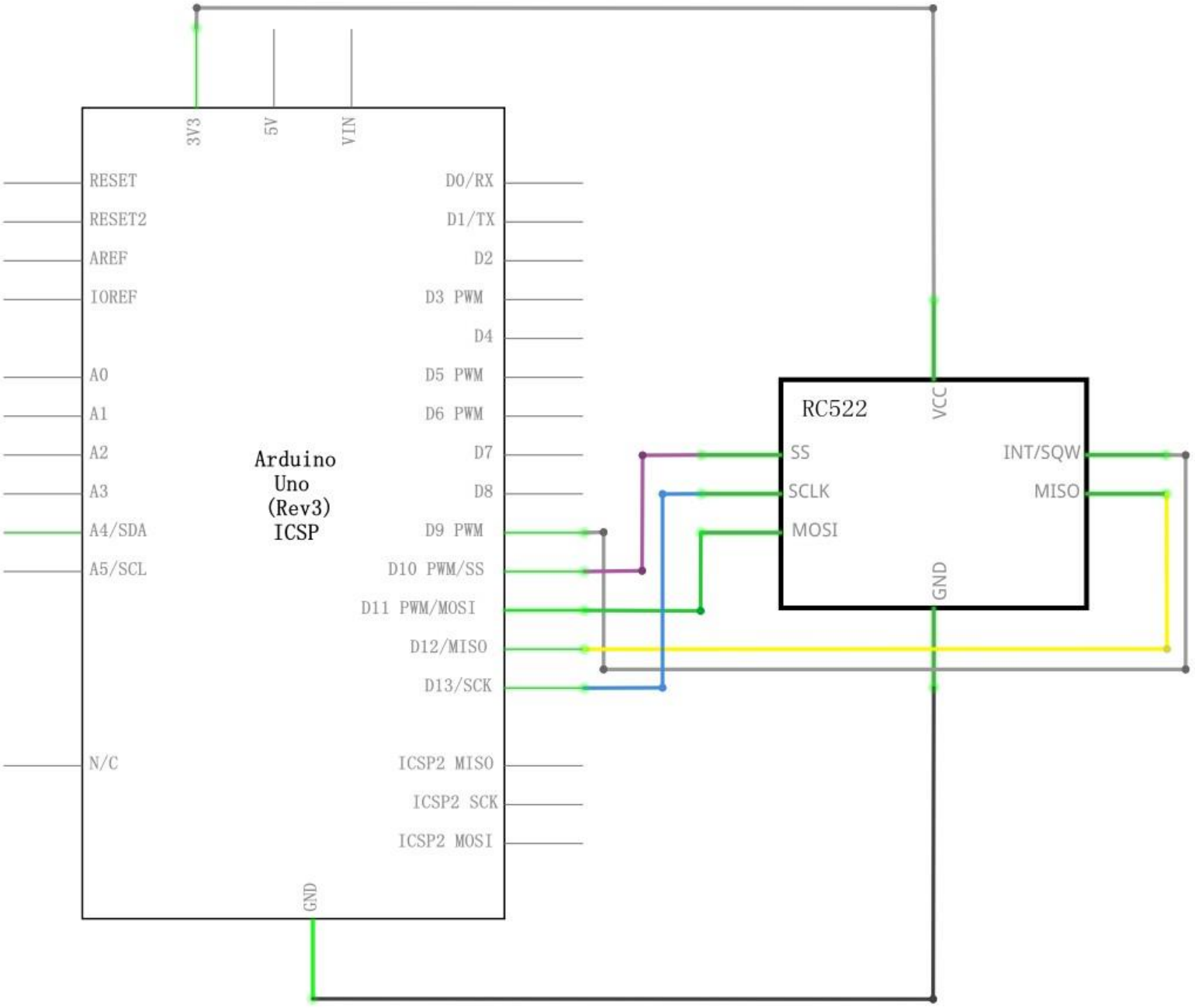
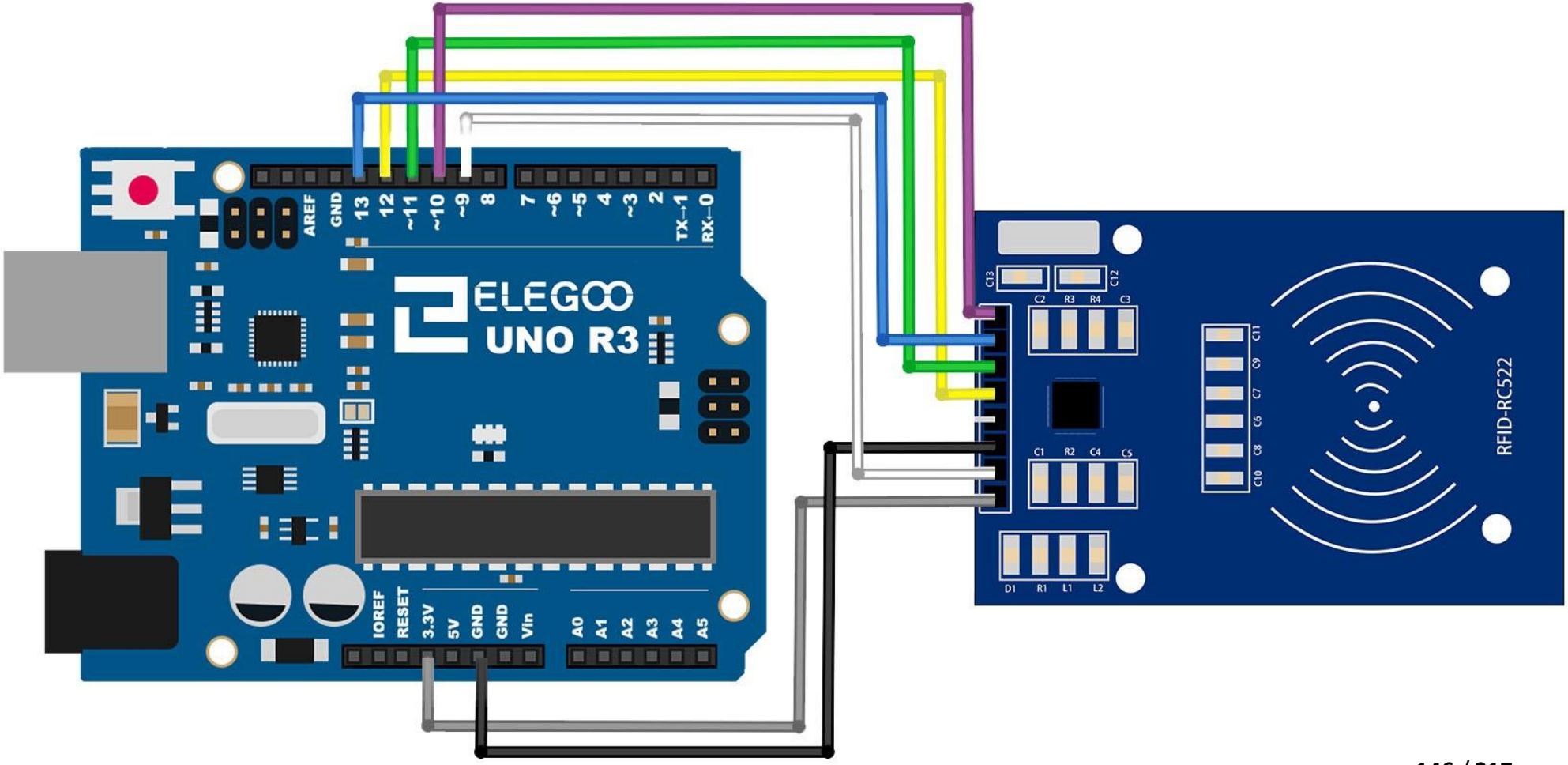


Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 21 RC522 RFID Module” et Téléversez-le sur la carte UNO.

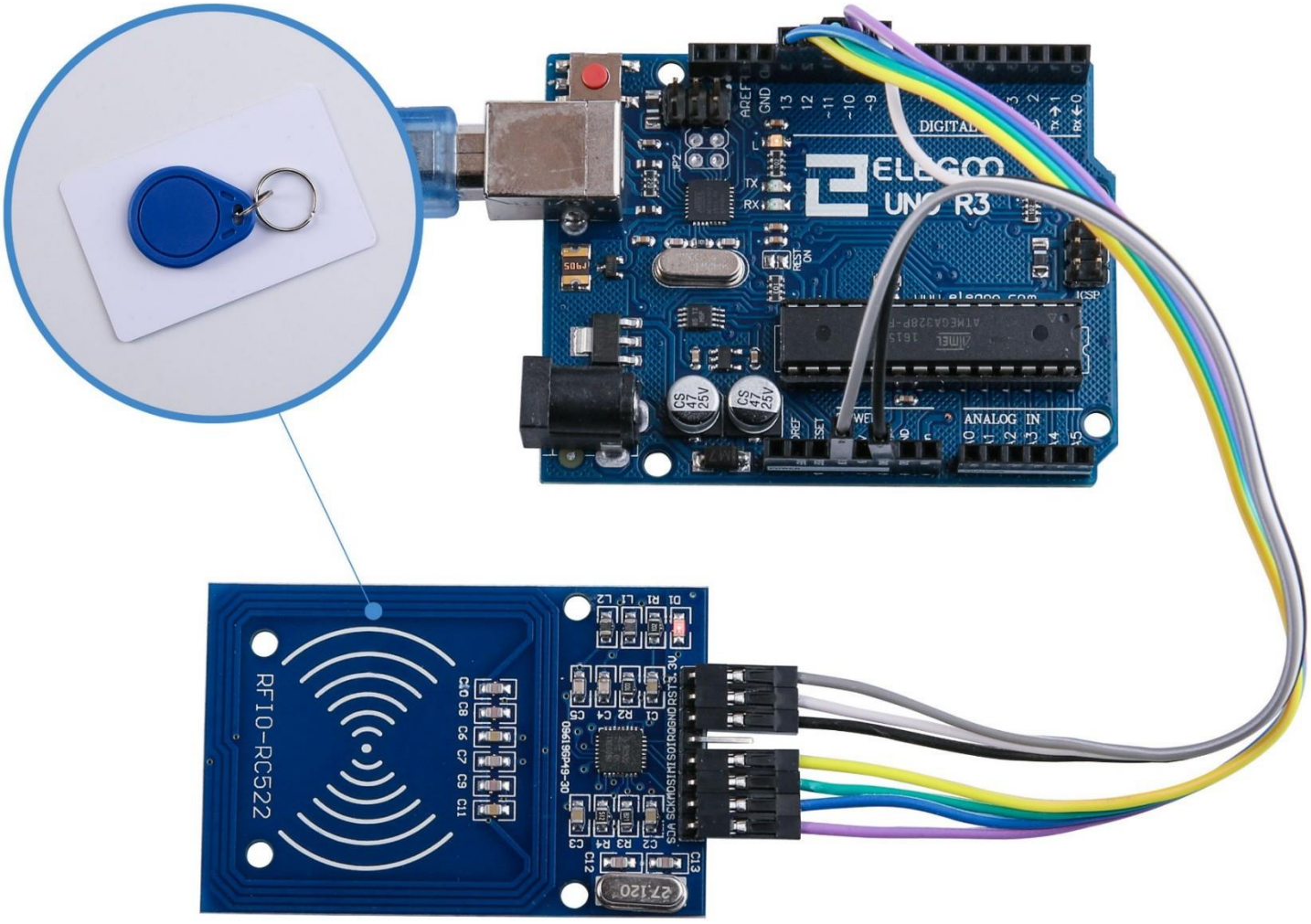
Avant toute chose, vérifiez la bonne installation de la bibliothèque < rfid >.

Signal	MFRC522 Reader/PCD Pin	Arduino Uno Pin	Arduino Mega Pin	Arduino Nano v3 Pin	Arduino Leonardo/Micro Pin	Arduino Pro Micro Pin
RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
SPI SS	SDA(SS)	10	53	D10	10	10
SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

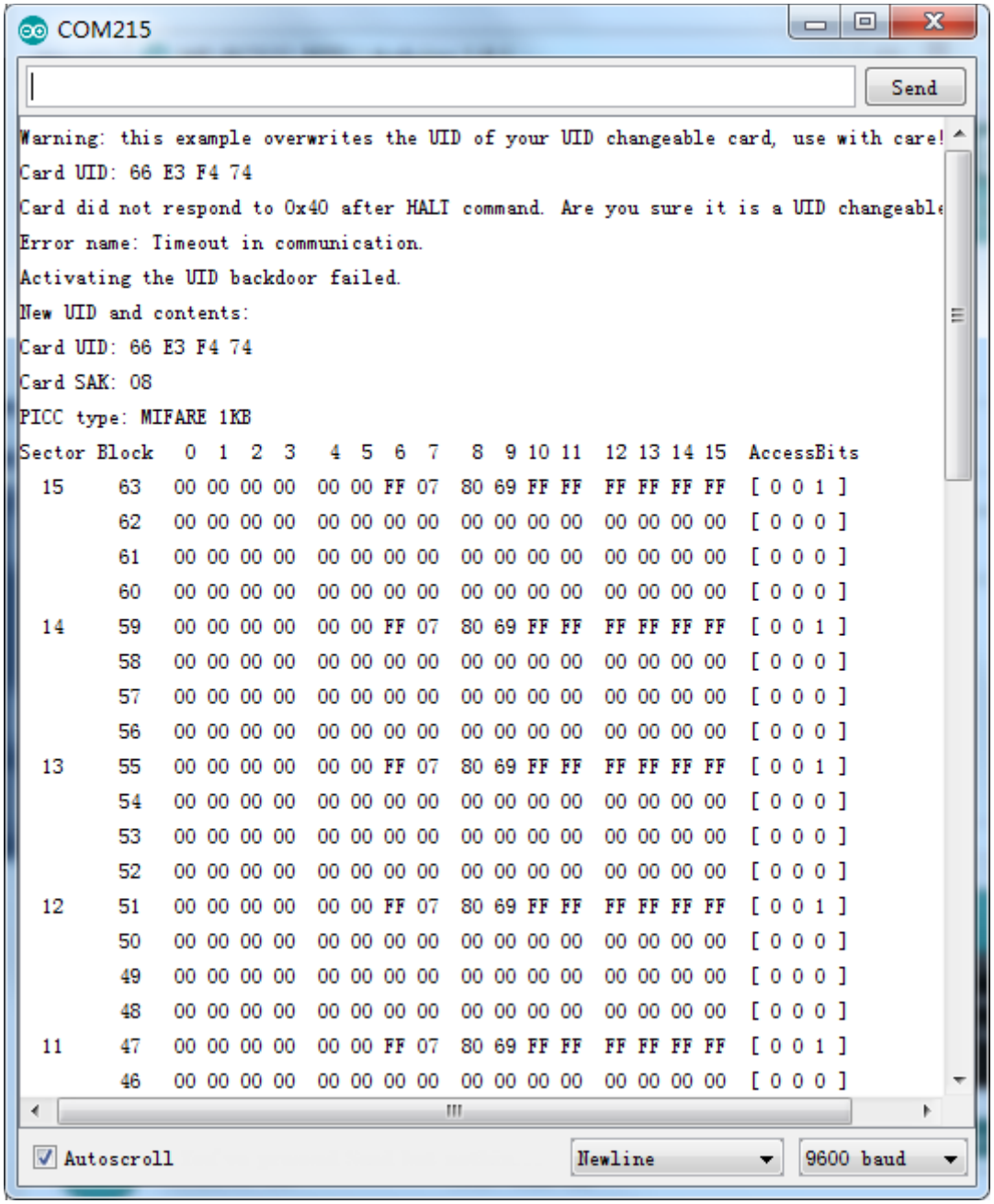
```
#define RST_PIN     9       // Configurable, see typical pin layout above
#define SS_PIN      10      // Configurable, see typical pin layout above
```

Attention, la position des pins peut varier en fonction des fournisseurs.

Illustration



Ouvrez le moniteur série et observez les données renvoyées par la carte:



Leçon 22 LCD Display

But de la leçon

Dans cette leçon, vous allez apprendre comment brancher et utiliser un afficheur alphanumérique.

L'afficheur dispose d'un rétro-éclairage LED et peut afficher deux lignes de 16 caractères. Chaque caractère est un rectangle de pixel. Il est possible de contrôler chaque pixel de chaque rectangle pour créer des caractères spécifiques.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Module LCD1602

(1) x Potentiomètre (10k)

(1) x Planche prototype

(16) x Câbles mâle-mâle

Présentation du composant

LCD1602

VSS: connecter à la masse

VDD: connecter au +5V

VO: connecter à un potentiomètre pour ajuster le contraste

RS: contrôler le registre de mémoire

R/W: sélectionner écriture ou lecture

E: Lorsqu'elle est à l'état bas, provoque l'exécution des instructions par le module LCD

D0-D7： Lire et Ecrire des données

A and K: Contrôler le rétro-éclairage

Connection

Schéma de câblage

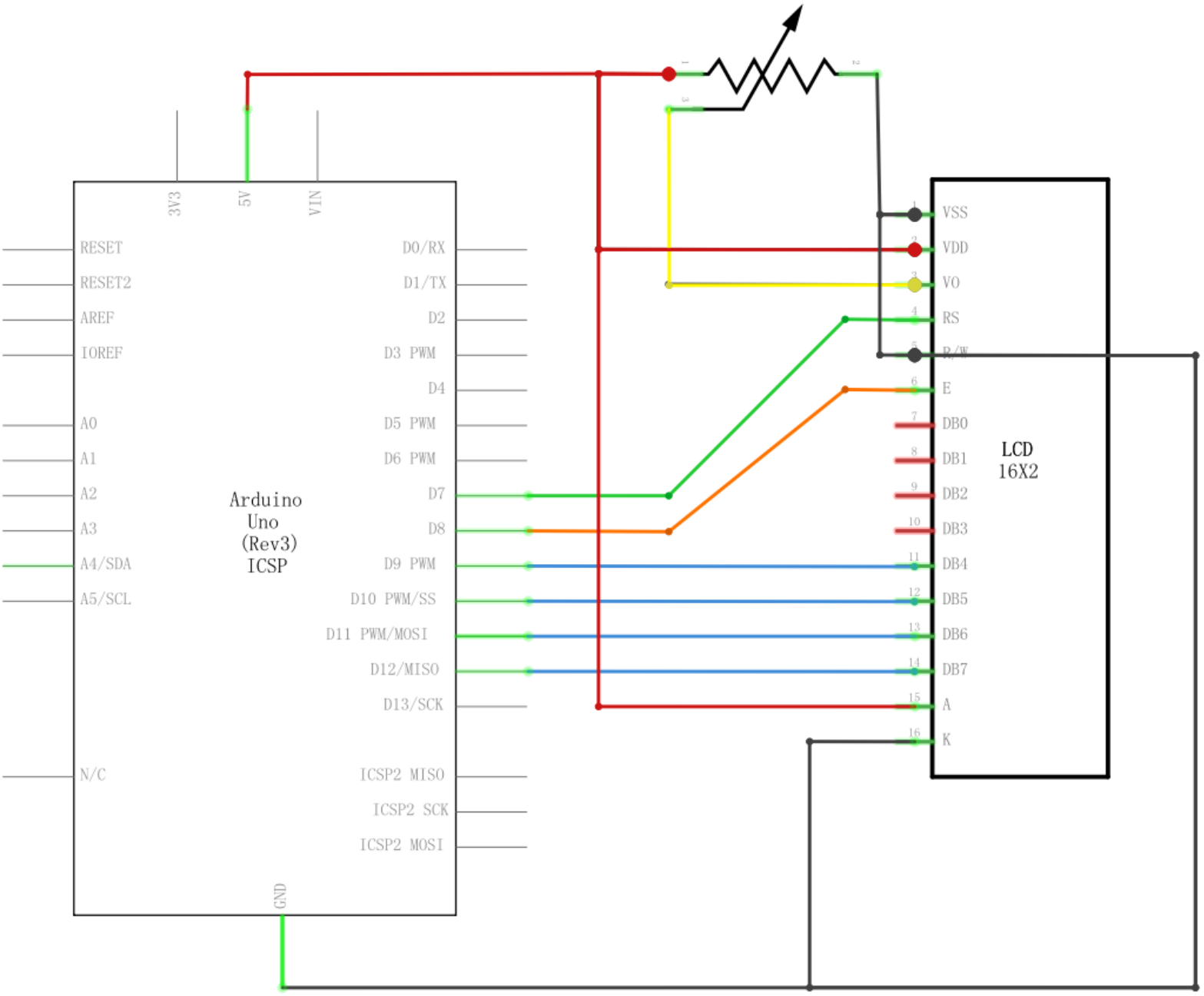
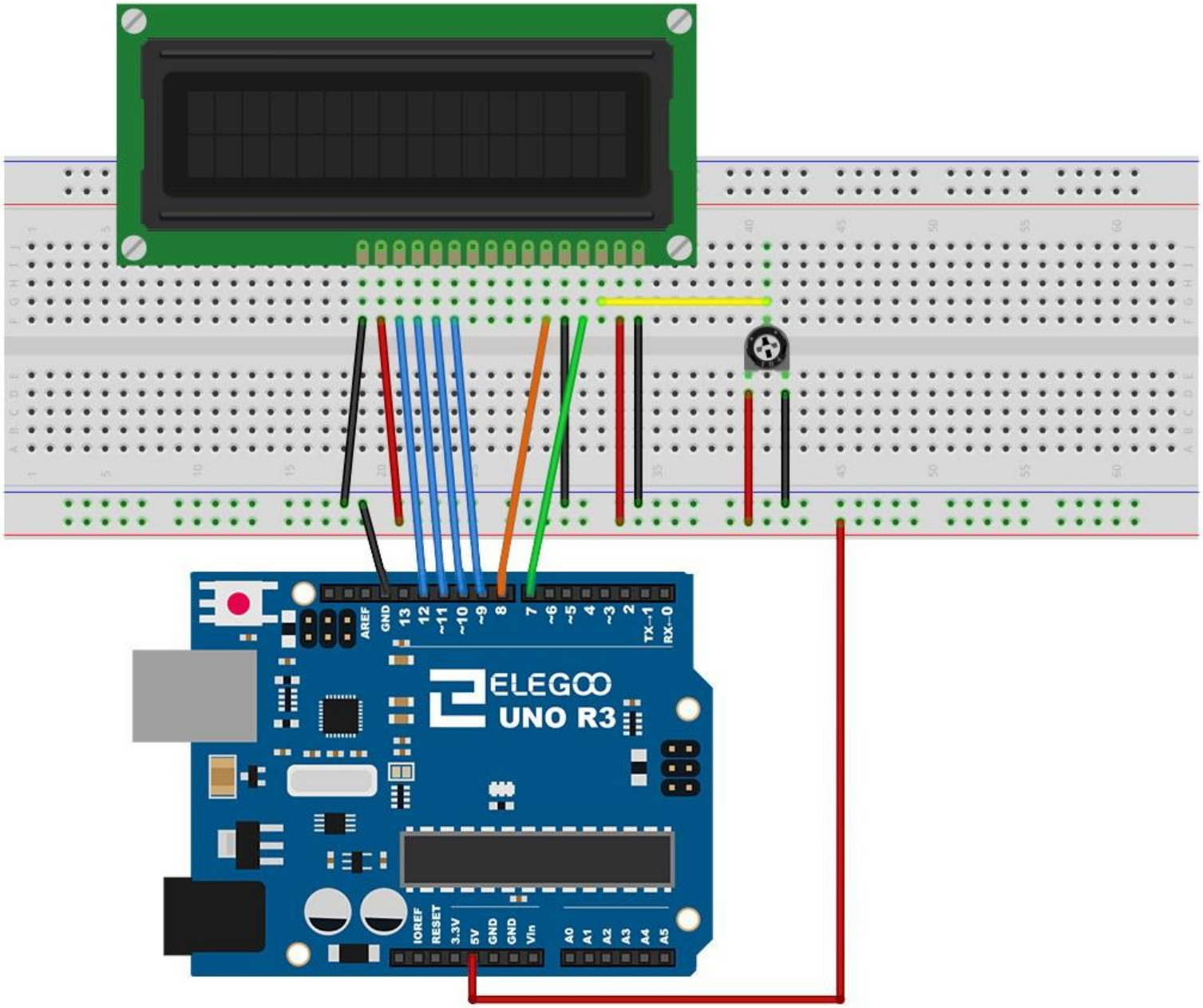


Diagramme de câblage



Comme vous pouvez le constater, le câblage est assez complexe. Lisez bien le plan et vérifier bien les connexions avant de faire la mise sous tension.
Un mauvais câblage ne doit pas aboutir à la destruction du module normalement, mais rien ne s'y affichera. Il faut donc procéder au montage avec le plus grand soin.

Code

Après avoir réalisé le câblage, ouvrez "Leçon 22 LCD Display" et Téléversez-le sur la carte UNO R3.
Vérifiez avant toute chose que la bibliothèque < LiquidCrystal > est bien installée.
Reportez-vous au nécessaire à la leçon 1 pour procéder à l'installation.

Après le téléversement, vous devriez voir apparaître le message « hello world » puis un comptage croissant partant de 0

Le sketch commence par la déclaration de la bibliothèque qui permet de piloter l'écran :

```
#include <LiquidCrystal.h>
```

La ligne suivante renseigne la bibliothèque sur les pins utilisées pour la connexion de l'écran :

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Les deux commandes suivantes ont pour effet de démarrer l'écran et d'y envoyer un message à afficher :

```
lcd.begin(16, 2);
```

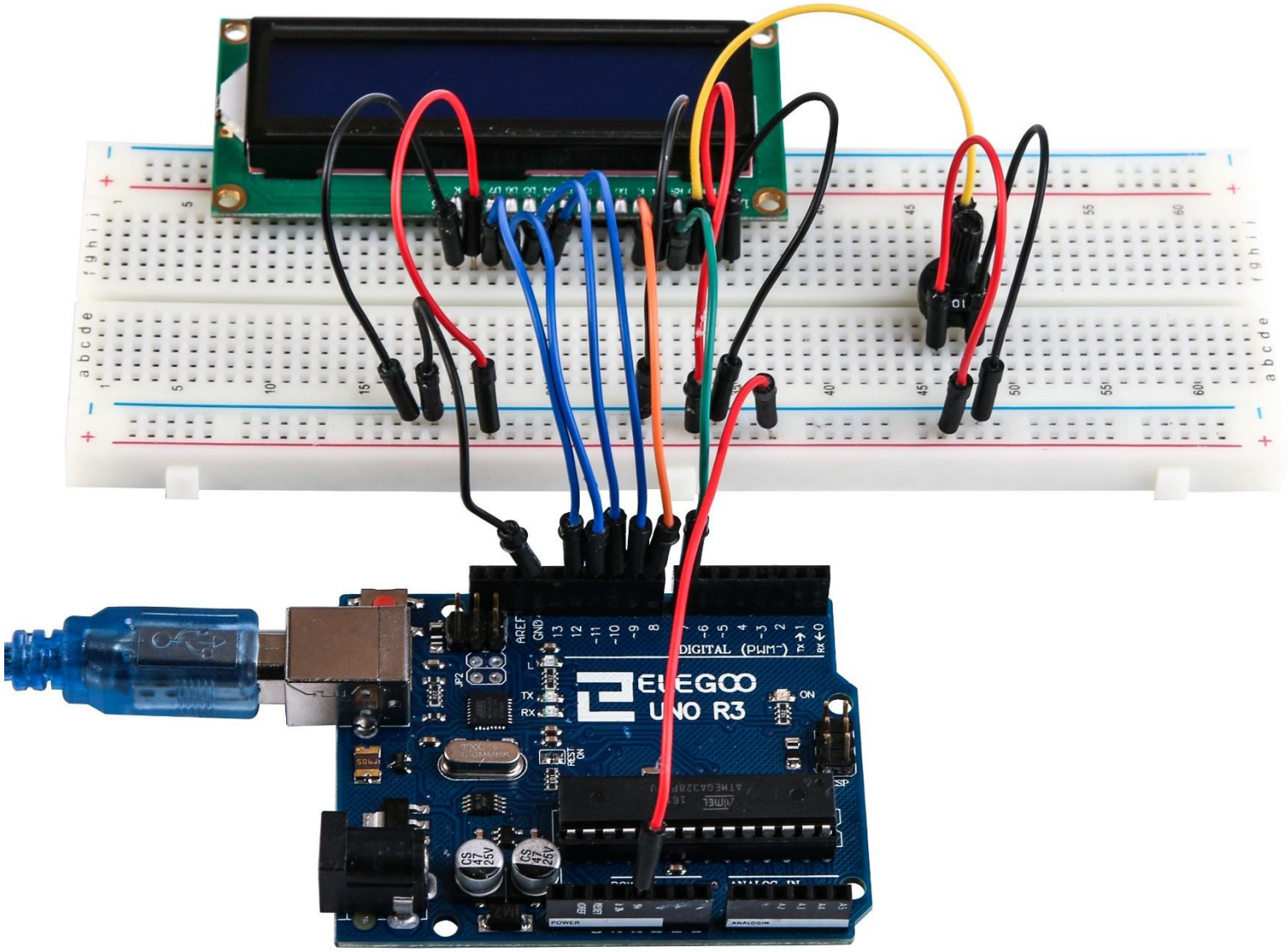
l'écran dispose de 16 colonnes et 2 lignes

```
lcd.print("Hello, World!");
```

Ces deux instructions démarrent le comptage.

```
lcd.setCursor(0, 1);
lcd.print(millis()/1000);
```

Illustration



Leçon 23 Thermometer

But de la leçon

Dans cette leçon, vous utiliserez un écran LCD pour afficher la température.

Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x Module LCD1602
(1) x 10k ohm resistor
(1) x Thermistance
(1) x Potentiomètre
(1) x Planche prototype
(18) x Câbles mâle-mâle

Présentation du composant

Thermistance

Une thermistance est une “résistance thermique” : sa valeur varie en fonction de la température. Une résistance standard voit aussi sa température varier, mais très faiblement, une thermistance est faite pour cette variation soit importante et donc facile à mesurer. (100 ohms par degré)

Il en existe deux types : NTC (negative temperature coefficient) et PTC (positive temperature coefficient). En général, on utilise des NTC.

Connection

Schéma de câblage

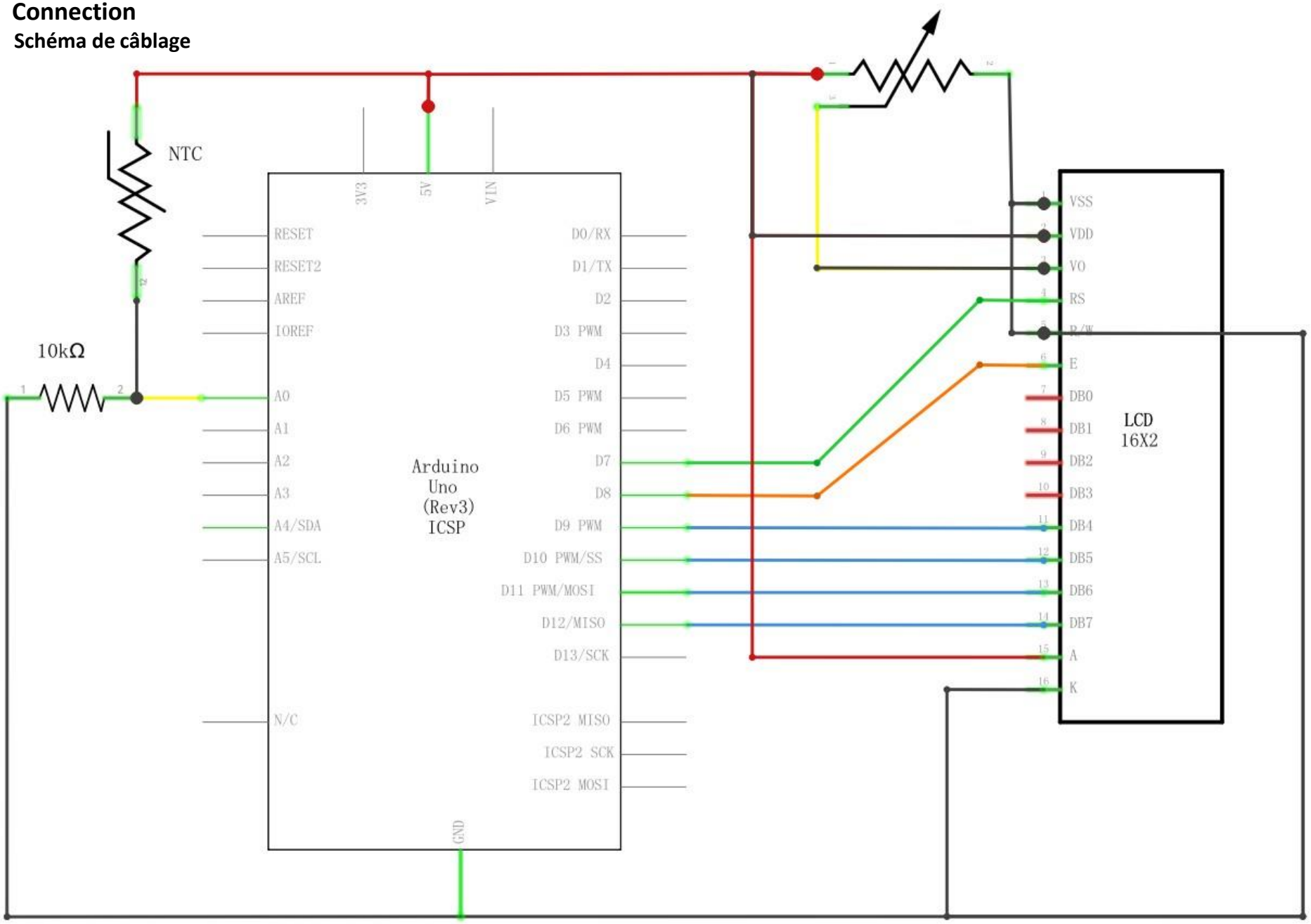
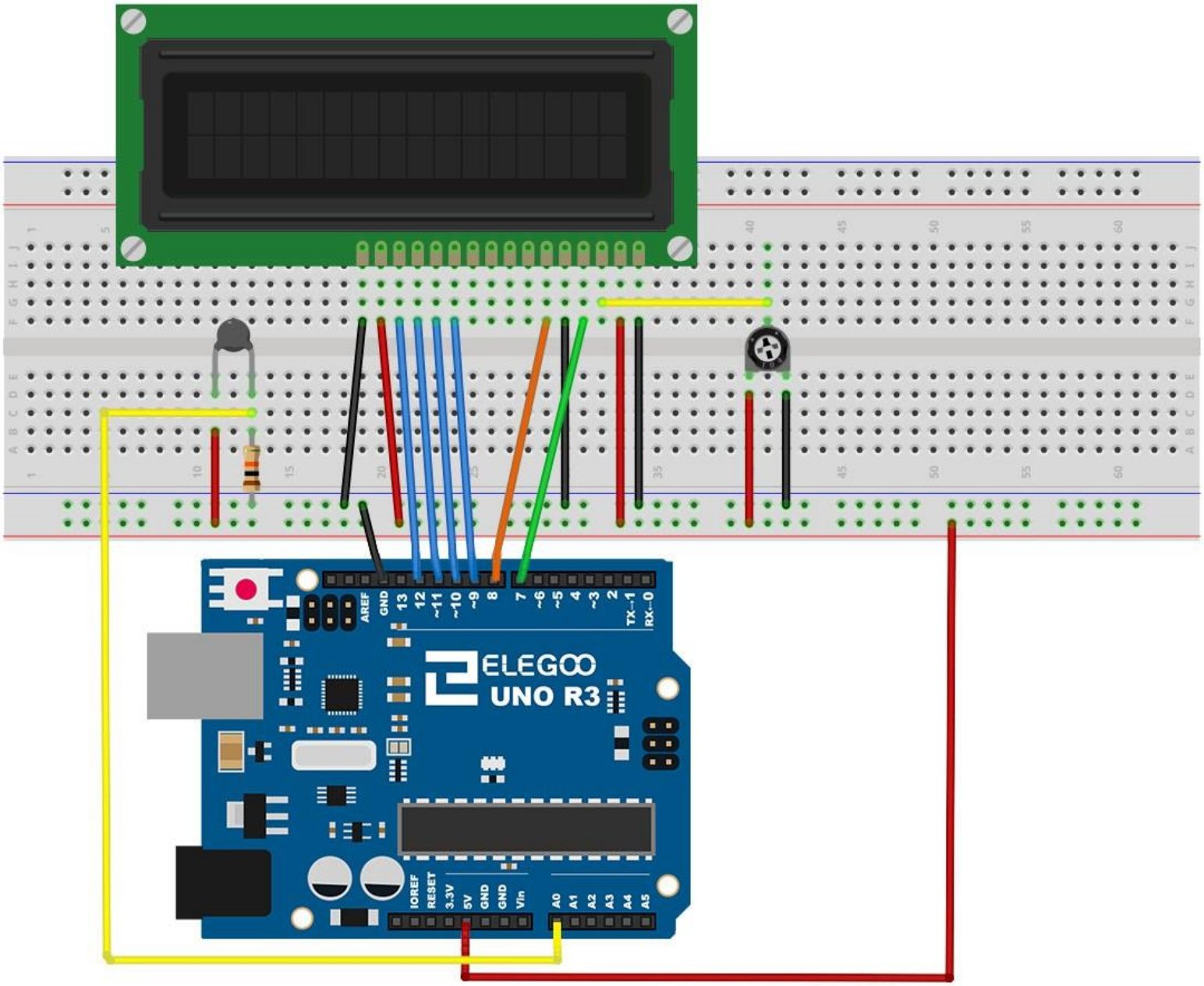


Diagramme de câblage



Comme vous pouvez le constater le schéma se base sur la leçon 22.

Code

Après avoir réalisé le câblage, ouvre le sketch “Leçon 23 Thermometer” et Téléversez-le sur la carte UNO R3.

Ce sketch est basé sur la leçon 22 et utilise la fonction “print” pour afficher la valeur de la température relevée au travers de la thermistance.

```
// BS   E   D4 D5   D6  D7
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Mais la valeur de la résistance, ne donne pas directement la température, il faut d’abord réaliser un calcul de conversion:

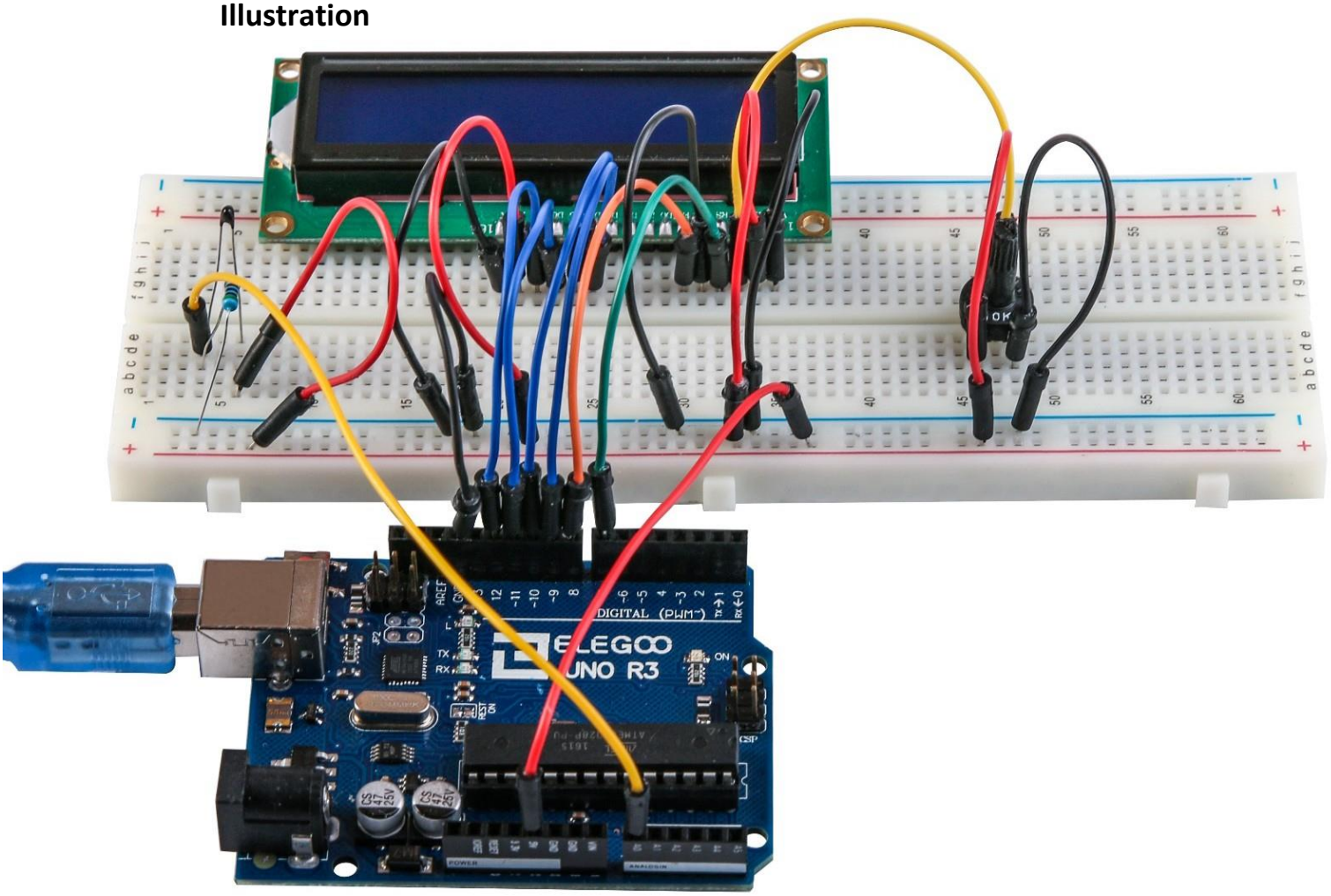
```
  int tempReading = analogRead(tempPin);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
 tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK ))
  * tempK );
  float tempC = tempK - 273.15;
  float tempF = (tempC * 9.0)/ 5.0 + 32.0;
```

Attention, changer une valeur mesurée sur un écran LCD peut être un peu délicat. En effet, il faut bien appréhender le nombre de digits utilisés. Par exemple, passer de 9 à 10 ou de 99 à 100 peut causer des faux affichages. Pour éviter cela, on procède à la mise à jour de l’ensemble de la ligne à chaque réaffichage.

```
lcd.setCursor(0, 0);
  lcd.print("Temp              C   ");
 lcd.setCursor(6, 0);
 lcd.print(tempF);
```

Observez l’image suivante et vous verrez comment la ligne complète est exploitée

Illustration



Leçon 24 Eight LED with 74HC595

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de commander indépendamment 8 leds sans avoir besoin de mobiliser 8 pins sur la carte UNO R3. Il est bien évidemment possible de connecter 8 leds sur 8 pins de votre carte UNO R3 si vous n'avez pas besoin de beaucoup d'autres connexions pour votre projet, cela fonctionnera parfaitement, mais il est vrai que la plupart du temps, il est nécessaire de brancher tout un tas de boutons, capteurs et donc les pins deviennent rapidement pas assez nombreuses. C'est pour cela que nous allons voir comment interfacer les leds à la carte via une puce 74HC595 qui est un convertisseur Série / Parallèle. Cette puce dispose de 3 entrée et 8 sorties.

Le recours à cette puce ralentira un peu les possibilités de switch on-off (de 800000 à 500000 par seconde), mais cela reste une fréquence plus que raisonnable.

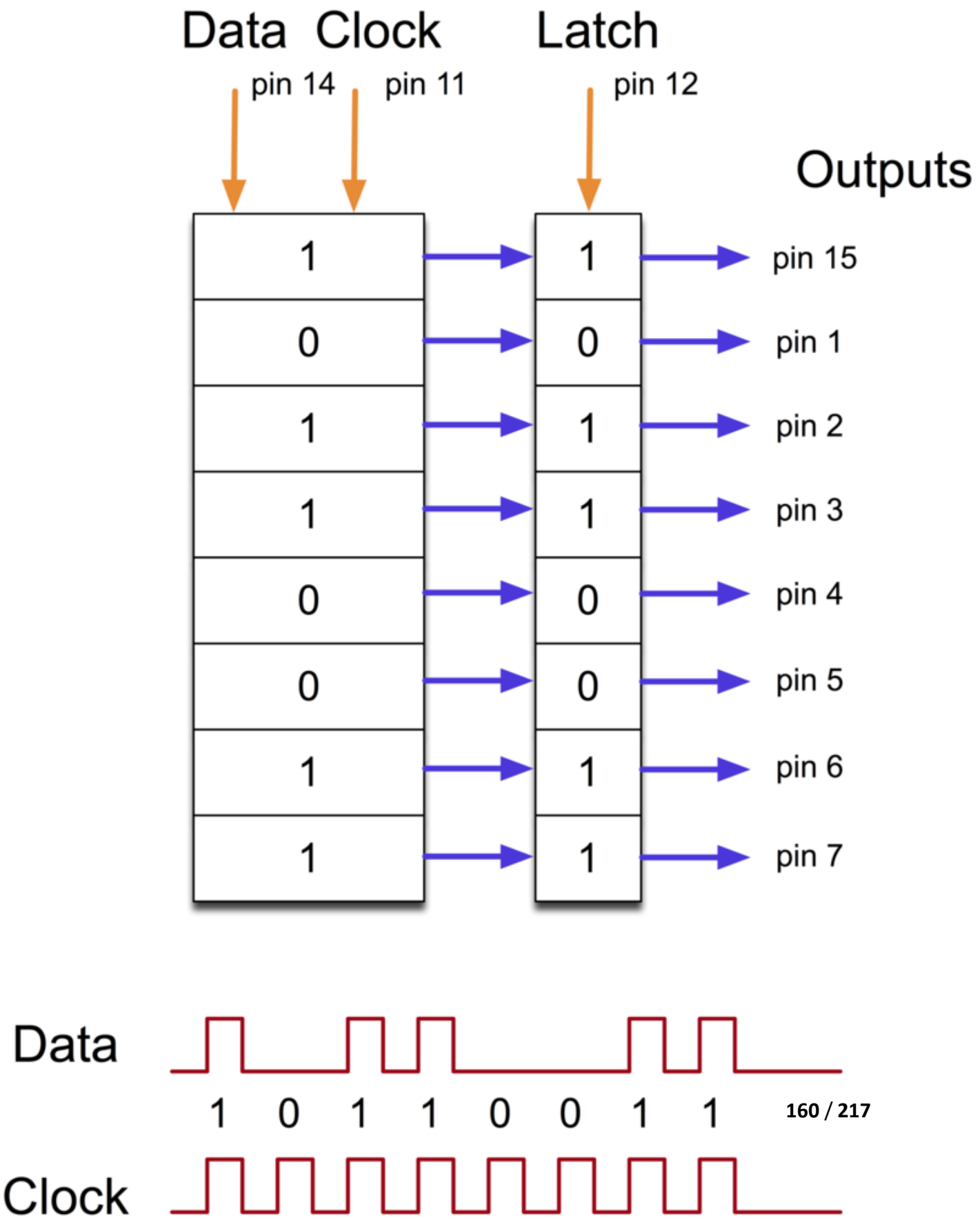
Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x planche prototype
(8) x leds
(8) x résistances 220 ohm
(1) x puce 74hc595
(14) x Câbles mâle-mâle

Présentation du composant

74HC595 Shift Register:

Le "shift register" est un type de puce qui contient un nombre de registres mémoire qui peuvent pendre soit la valeur 0, soit la valeur 1. Il est possible de changer chaque valeur indépendamment.


Data
Clock
Latch
pin 14
pin 11
pin 12
Outputs
1
0
1
1
0
0
1
1
1
0
1
1
0
0
1
1
pin 15
pin 1
pin 2
pin 3
pin 4
pin 5
pin 6
pin 7
Data
1 0 1 1 0 0 1 1
Clock

Pour cette puce, l'horloge fonctionne avec 8 pulsations. A chaque pulsation correspond une adresse registre. Si la pin est à l'état haut, le registre prendra la valeur 1, s'il est à l'état bas, il prendra la valeur 0.

Connection

Schéma de câblage

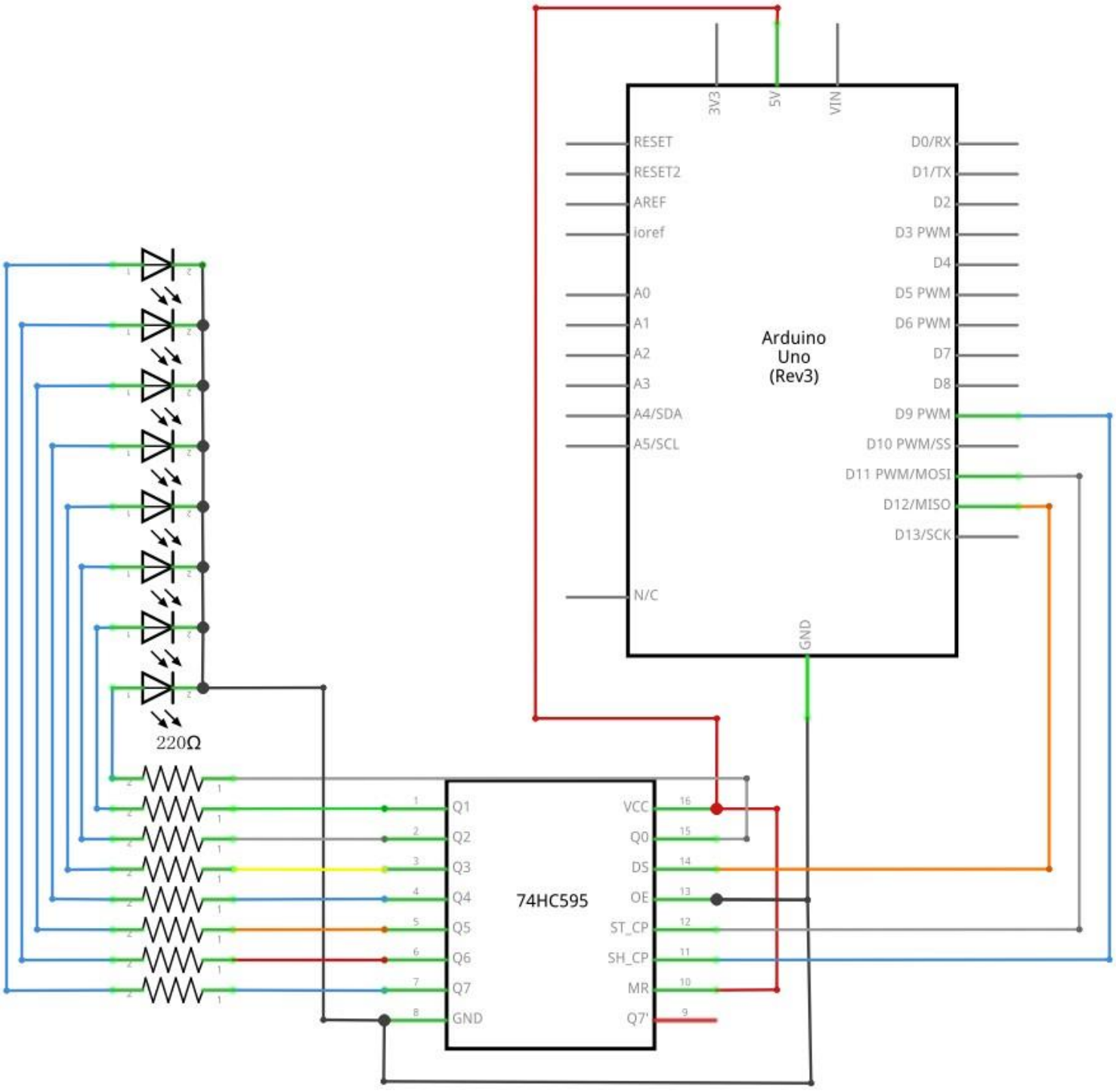
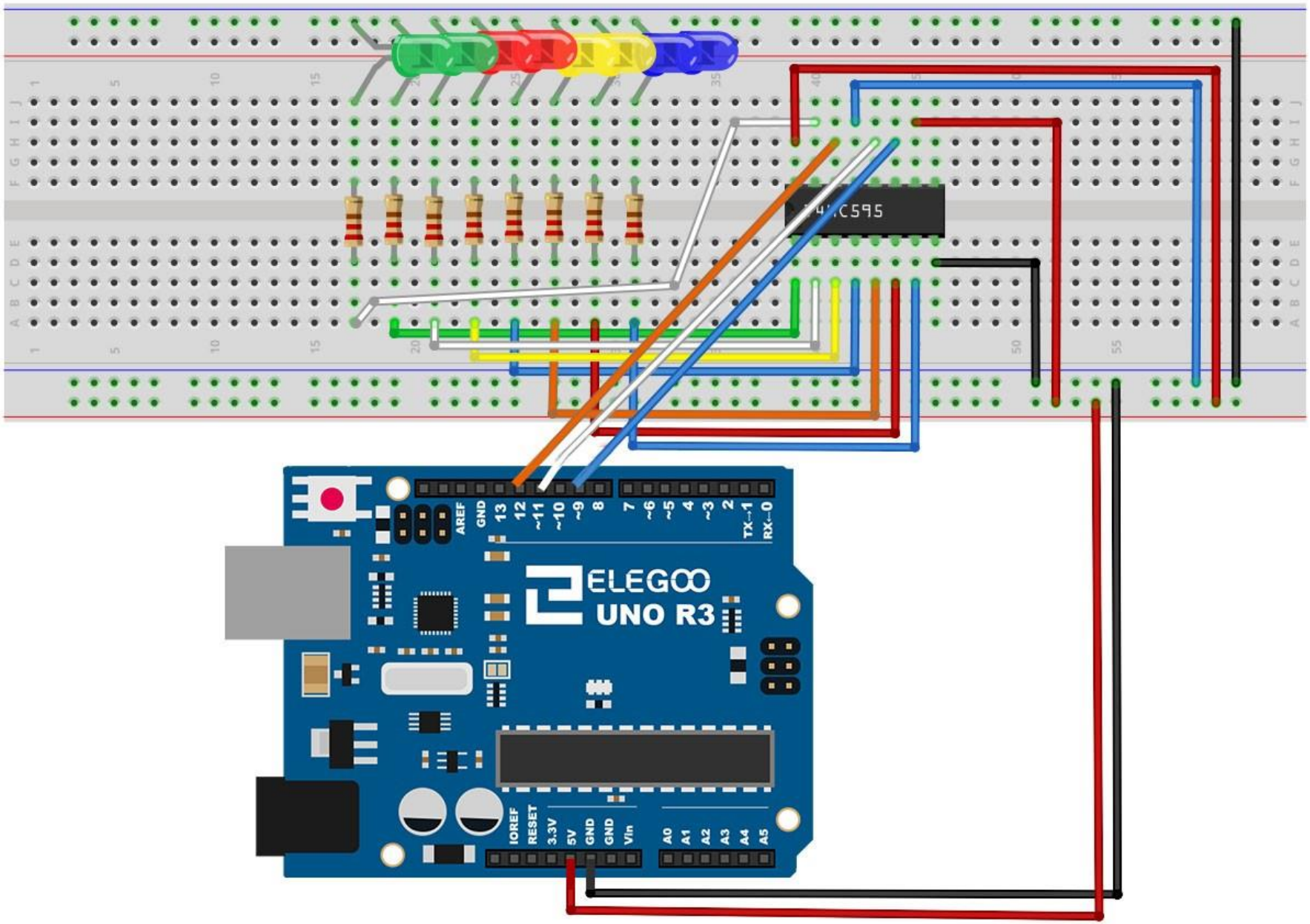


Diagramme de câblage



Ce schéma est assez complexe et demande d'être fait avec soin.
Le mieux étant de commencer par la mise en place de la puce 74HC595.
Toutes les leds sauf une sont sur le même côté de la puce.
Après avoir mis la puce en place installez les résistances.
Placez ensuite les différentes LEDS. Faites bien attention au sens des leds, la patte la plus longue devant être du côté de la borne positive du montage.

Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon 24 Eight LED with 74HC595" et Téléversez-le sur la carte UNO R3.

Les premières lignes du code définissent les pins de la carte UNO R3

```
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
```

Ensuite, une variable de type 'byte' est définie pour la définition de quelle led est à l'état allumée et quelle est à l'état éteinte.

```
byte leds = 0;
```

La function 'setup' réalise l'affectation des pins sur la carte

```
void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
```

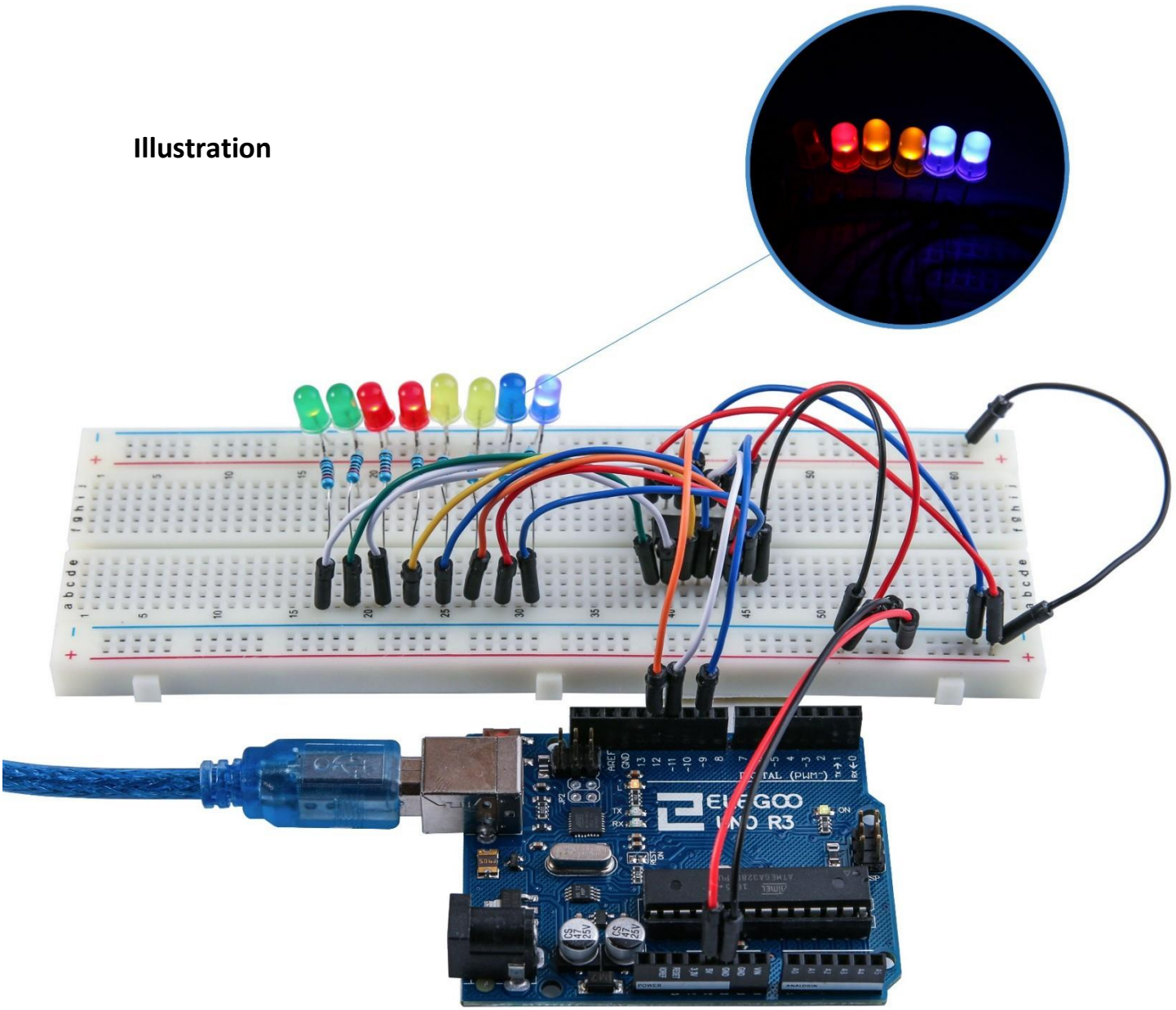
La fonction loop commence par définir toutes les leds à 'éteinte' en donnant la valeur 0 à la variable 'led'. Ensuite, elle appelle ensuite 'updateShiftRegister' qui va permettre l'allumage de certaines leds et l'extinction d'autres en définissant les valeurs 0 ou 1.
La fonction loop fait ensuite une pause de 500ms et procède ensuite à l'allumage/extinction des leds.

```
void loop()
{
   leds = 0;
   updateShiftRegister();
   delay(500);
   for (int i = 0; i < 8; i++)
   {
      bitSet(leds, i);
      updateShiftRegister();
      delay(500);
   }
}
```

La fonction 'updateShiftRegister', commence par définir latchPin à bas, puis appelle la fonction UNO 'shiftOut' avant de remettre 'latchPin' à haut.

```
void updateShiftRegister()
{
     digitalWrite(latchPin, LOW);
     shiftOut(dataPin, clockPin, LSBFIRST, leds);
     digitalWrite(latchPin, HIGH);
}
```

Illustration



Leçon 25 The Serial Monitor

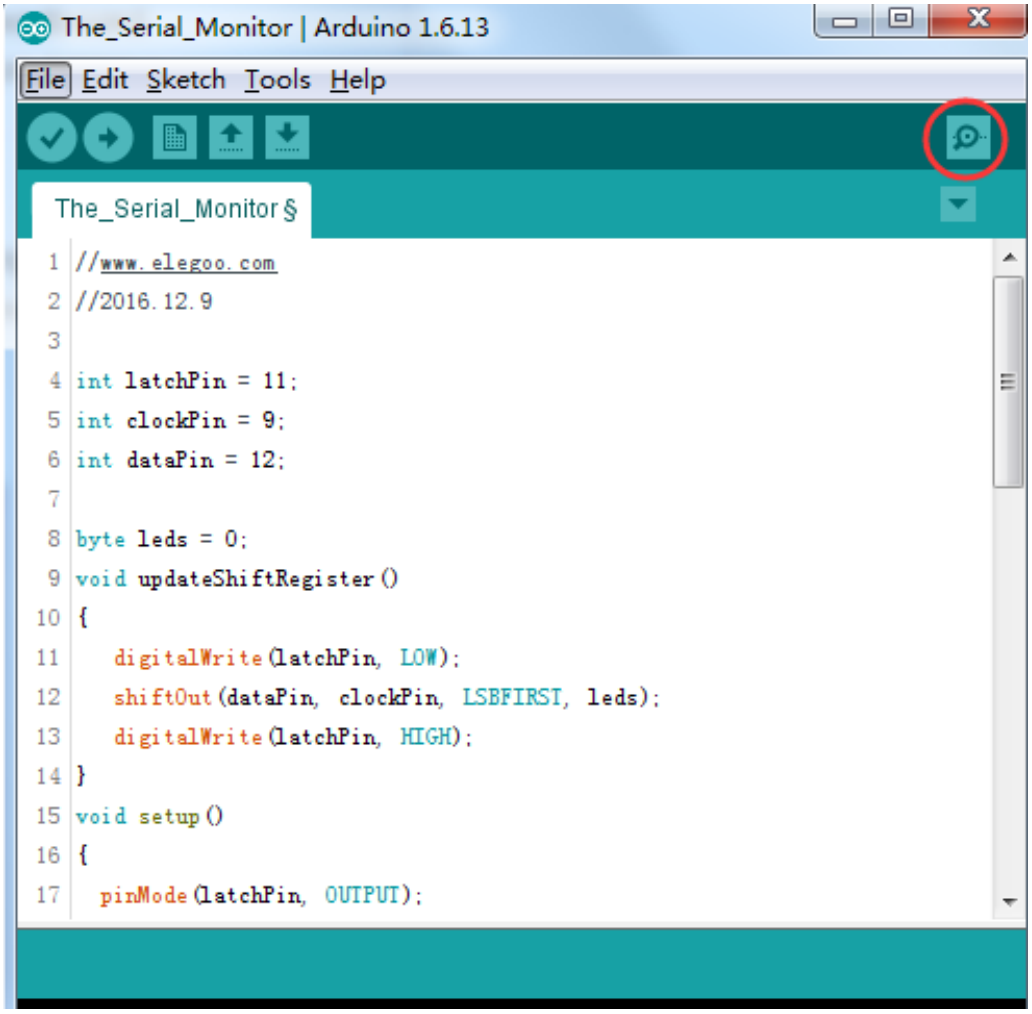
But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de contrôler l'allumage/extinction des leds avec le moniteur série. Cette leçon repose sur la leçon 24 qu'il est impératif d'avoir accompli avant.

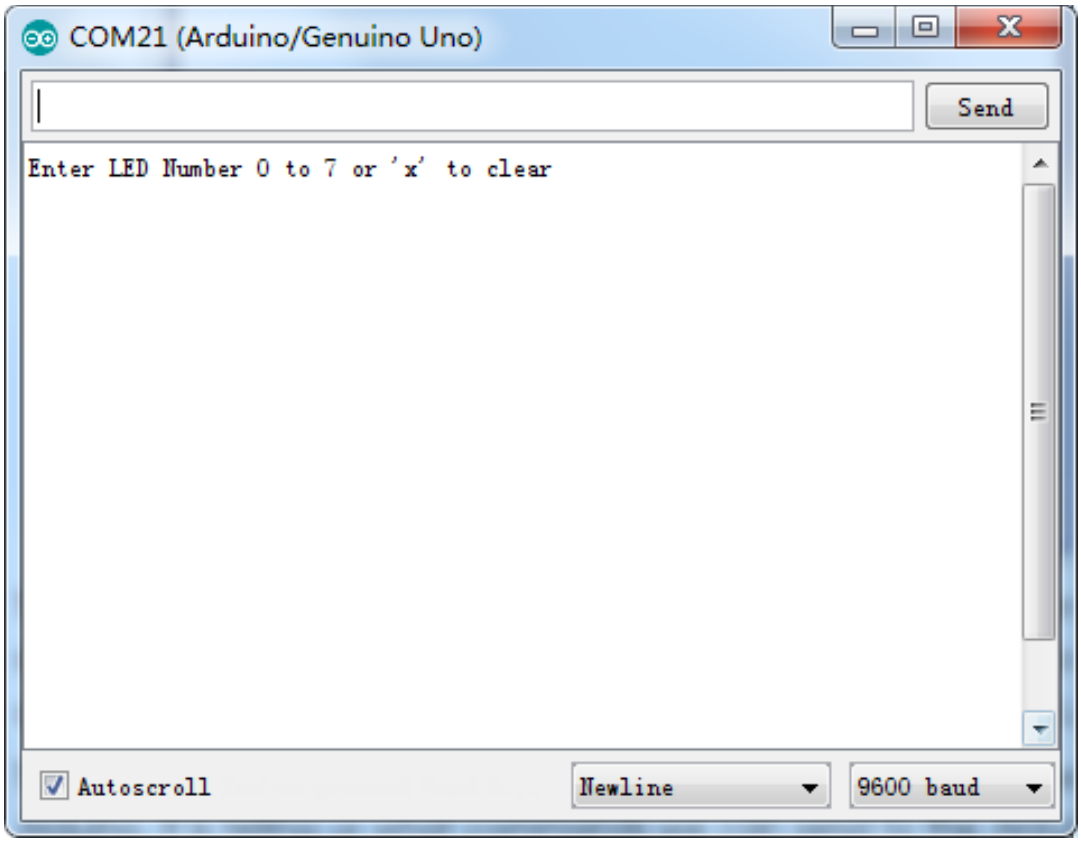
Dans cette leçon, le montage électronique est strictement identique à la leçon 24.

Etapes

Après avoir ouvert et téléversé le sketch, ouvrez le moniteur série.

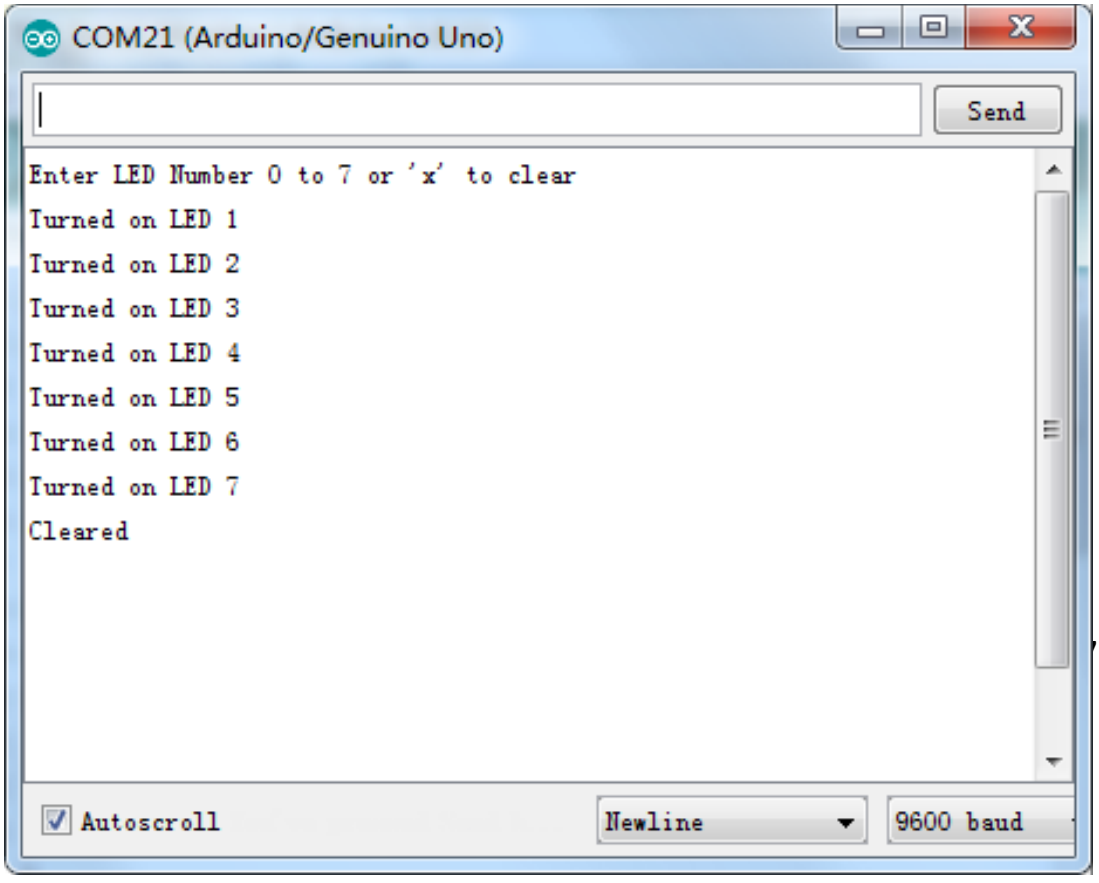


Voici ce que vous devez observer:



Vous êtes invités à entrer un chiffre ou la lettre *x* puis valider.

Envoyer un chiffre allumera la led correspondante, envoyer x les éteindra toutes.



Code

Ouvrez le sketch “Leçon 25 The Serial Monitor” et Téléversez-le sur la carte UNO R3. Comme vous pouvez vous y attendre, le code est très proche du précédent. Nous allons simplement nous attarder sur les instructions supplémentaires.

```
void setup()
{
   pinMode(latchPin, OUTPUT);
   pinMode(dataPin, OUTPUT);
   pinMode(clockPin, OUTPUT);
   updateShiftRegister();
   Serial.begin(9600);
   while (! Serial); // Wait until Serial is ready - Leonardo
   Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

Tout d'abord, nous avons la commande 'Serial.begin (9600)'. Cela initialise la communication série, de sorte que la carte UNO R3 peut envoyer des commandes via la connexion USB. La valeur 9600 est appelée «débit en bauds» de la connexion. Cela correspond à la vitesse d’échange entre l’ordinateur et la carte.

La ligne commençant par 'while' garantit qu'il y a quelque chose à l'autre extrémité de la connexion USB pour l'Arduino avant de commencer à envoyer des messages. Sinon, le message peut être envoyé, mais pas affiché. Cette ligne n'est réellement nécessaire que si vous utilisez une carte Leonardo car la carte UNO R3 réinitialise automatiquement la carte lorsque vous ouvrez le moniteur série, alors que cela ne se produit pas avec la Leonardo.

La dernière des nouvelles lignes dans 'setup' envoie le message que nous voyons en haut du moniteur série.

```
void loop()
{
   if (Serial.available())
   {
      char ch = Serial.read();
      if (ch >= '0' && ch <= '7')
```

```
      {
         int led = ch - '0';
         bitSet(leds, led);
         updateShiftRegister();
         Serial.print("Turned on LED ");
         Serial.println(led);
      }
      if (ch == 'x')
      {
         leds = 0;
         updateShiftRegister();
         Serial.println("Cleared");
      }
   }
}
```

Tout ce qui se passe dans la boucle se trouve dans une déclaration «if». Donc, à moins que l'appel à la fonction Arduino intégrée 'Serial.available ()' soit 'vrai', rien d'autre ne se produira.

Serial.available () renverra 'true' si les données ont été envoyées à l'UNO et sont prêtes à être traitées. Les messages entrants sont contenus dans ce qu'on appelle un tampon et Serial.available () renvoie true si ce buffer n'est pas vide.

Si un message a été reçu, il est sur la ligne de code suivante:

```
char ch = Serial.read();
```

Cette instruction lit le caractère suivant du tampon et le supprime du tampon. Il l'affecte également à la variable 'ch'. La variable 'ch' est du type 'char' qui signifie 'caractère' et, comme son nom l'indique, détient un caractère unique.

Si vous avez suivi les instructions dans l'invite située en haut du Serial Monitor, ce caractère sera soit un chiffre d'un seul chiffre compris entre 0 et 7, soit la lettre «x». L'instruction 'if' sur la ligne suivante vérifie si c'est un seul chiffre en voyant si 'ch' est supérieur ou égal au caractère '0' et inférieur ou égal au caractère '7'. Il semble étrange de comparer des variables de cette façon, mais c'est parfaitement acceptable.

Chaque caractère est représentée par un nombre unique, appelé sa valeur ASCII.

Cela signifie que lorsque nous comparons des caractères en utilisant <= et> = c'est en fait les valeurs ASCII qui ont été comparées.

Si le test passe, nous arrivons à la ligne suivante:

```
int led = ch – '0';
```

Maintenant, nous interprétons l'arithmétique sur les charactères! Nous soustrayons le chiffre '0' de n'importe quel chiffre saisi. Donc, si vous avez tapé '0', alors '0' - '0' sera égal à 0. Si vous avez tapé '7', alors '7' - '0' sera égal au nombre 7 car il s'agit en fait des valeurs ASCII utilisées dans la soustraction.

Puisque nous connaissons le nombre de LED que nous voulons allumer, il suffit de définir ce bit dans la variable 'leds' et de mettre à jour le registre à décalage.

```
bitSet(leds, led);
updateShiftRegister();
Serial.print("Turned on LED ");
Serial.println(led);
```

La première ligne utilise Serial.print plutôt que Serial.println. La différence entre les deux est que Serial.print ne démarre pas une nouvelle ligne après l'affichage, ce qui est dans son paramètre. Nous utilisons cela en première ligne, car nous affichons le message en deux parties. Tout d'abord, le libellé général: «LED allumée», puis le nombre de LED.

Le nombre de LED est maintenu dans une variable 'int' plutôt que d'être une chaîne de texte. Serial.print peut prendre soit une chaîne de texte entre guillemets doubles, soit un «int» ou, en fait, pratiquement n'importe quel type de variable.

Après la déclaration «if» qui gère le cas, lorsqu'un seul chiffre a été géré, il y a une deuxième instruction «if» qui vérifie si «ch» est la lettre «x».

```
if (ch == 'x')
    {
      leds = 0;
      updateShiftRegister();
      Serial.println("Cleared");
    }
```

Leçon 26 Photocell

But de la leçon

Dans cette leçon, vous allez apprendre à mesurer l'intensité de la lumière en utilisant une entrée analogique. Vous allez utiliser le montage réalisé à la leçon précédente et utiliser vos nouvelles connaissances pour contrôler le nombre de leds allumées.

Matériel nécessaire:

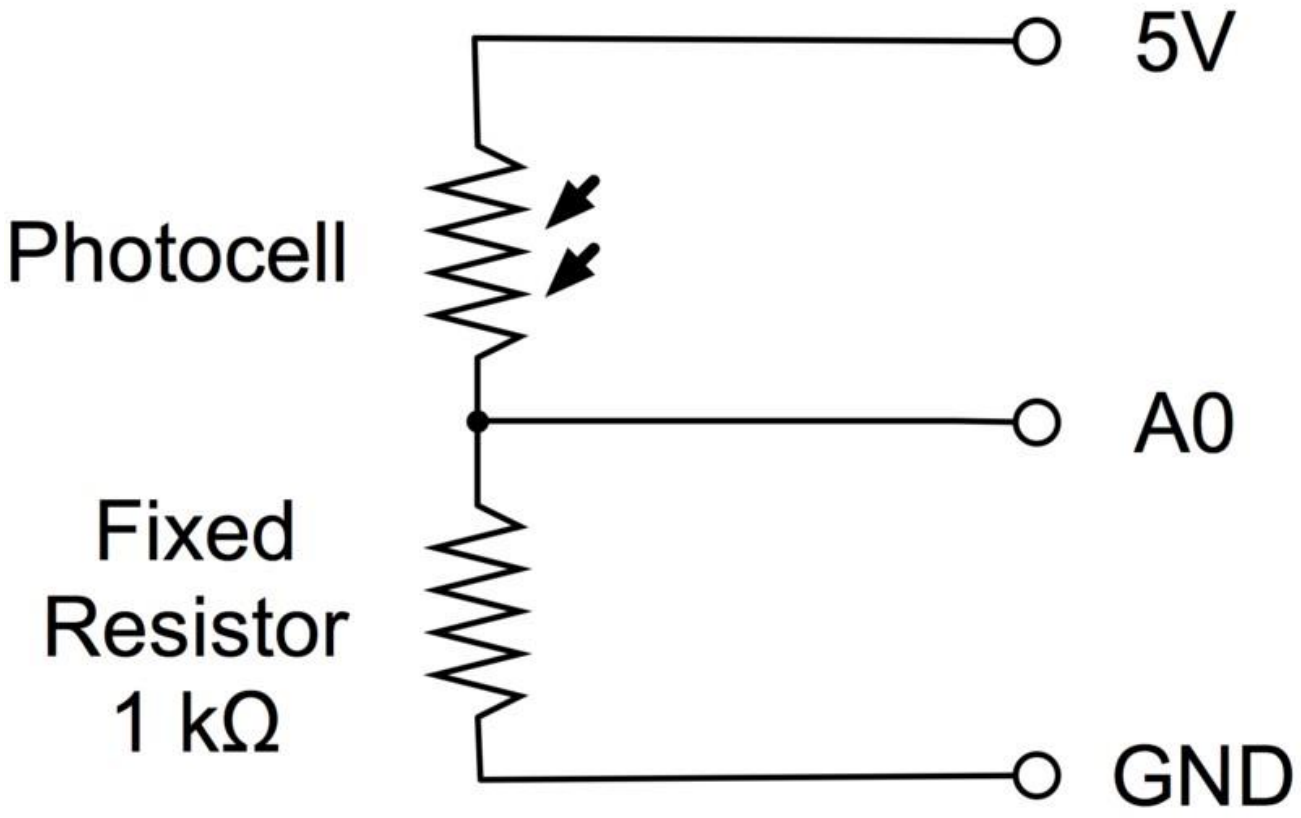
(1) x Elegoo Uno R3
(1) x Planche prototype
(8) x Leds
(8) x Résis220 ohm resistors
(1) x 1k ohm resistor
(1) x 74hc595 IC
(1) x Photorésistance (Photocell)
(16) x Câbles mâle-mâle

Présentation du composant

PHOTOCELL:

La photocell utilisée est appelée "résistance dépendante de la lumière" ou light dependent resistor (LDR). Comme son nom l'indique la valeur de la résistance dépend de la quantité de lumière reçue par le composant.

Cette résistance prend une valeur de 50 kΩ en obscurité et varie jusqu'à 500 Ω en pleine lumière. Pour convertir cette variation de valeur en variation de courant, il faut bien entendu alimenter la résistance.


5V
Photocell
A0
Fixed
Resistor
1 kΩ
GND

Connection

Schéma de câblage

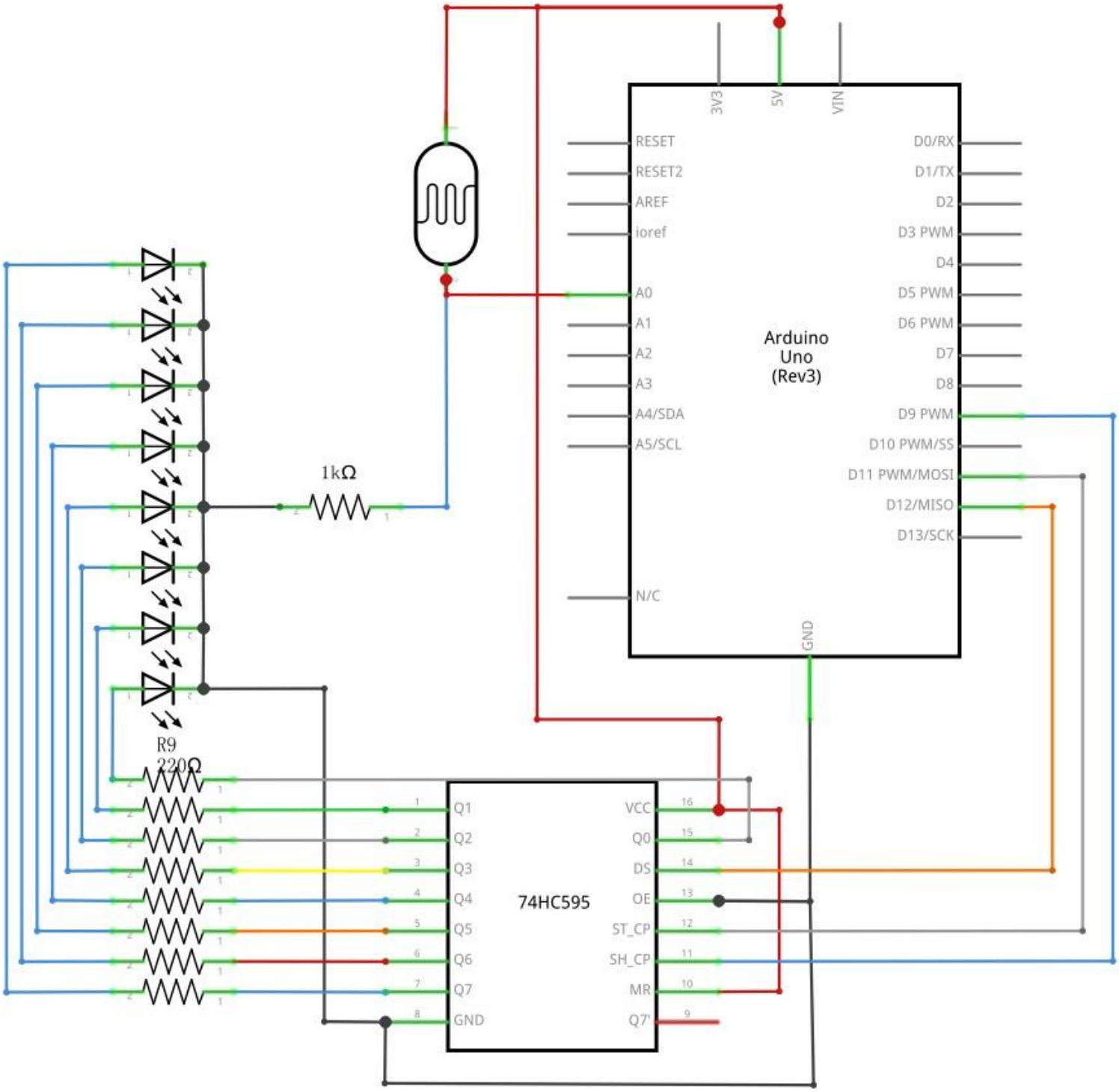
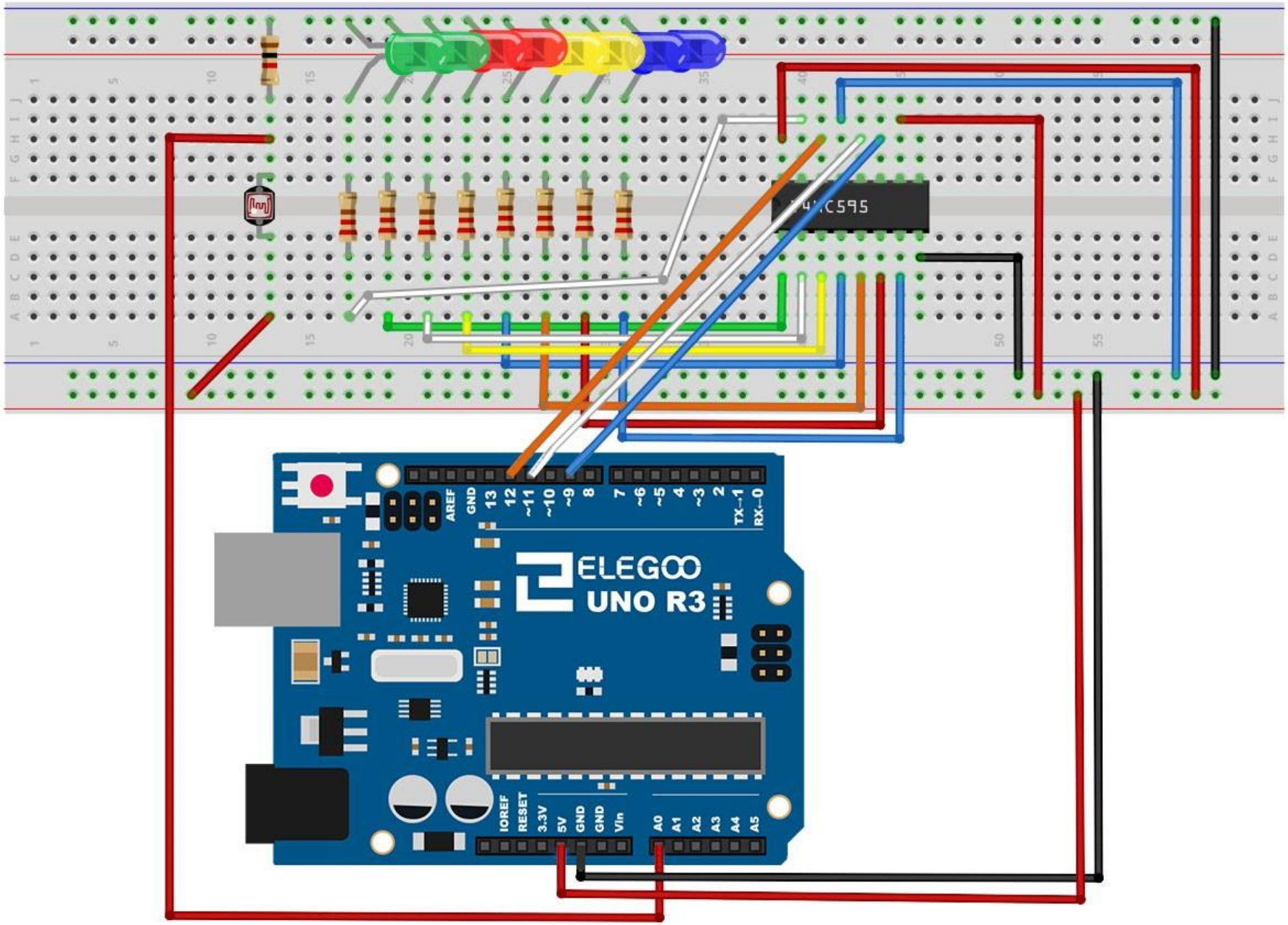


Diagramme de câblage



Code

Après avoir réalisé le schéma de câblage, ouvrez le sketch “Leçon 26 Photocell” et Téléversez-le sur la carte UNO.

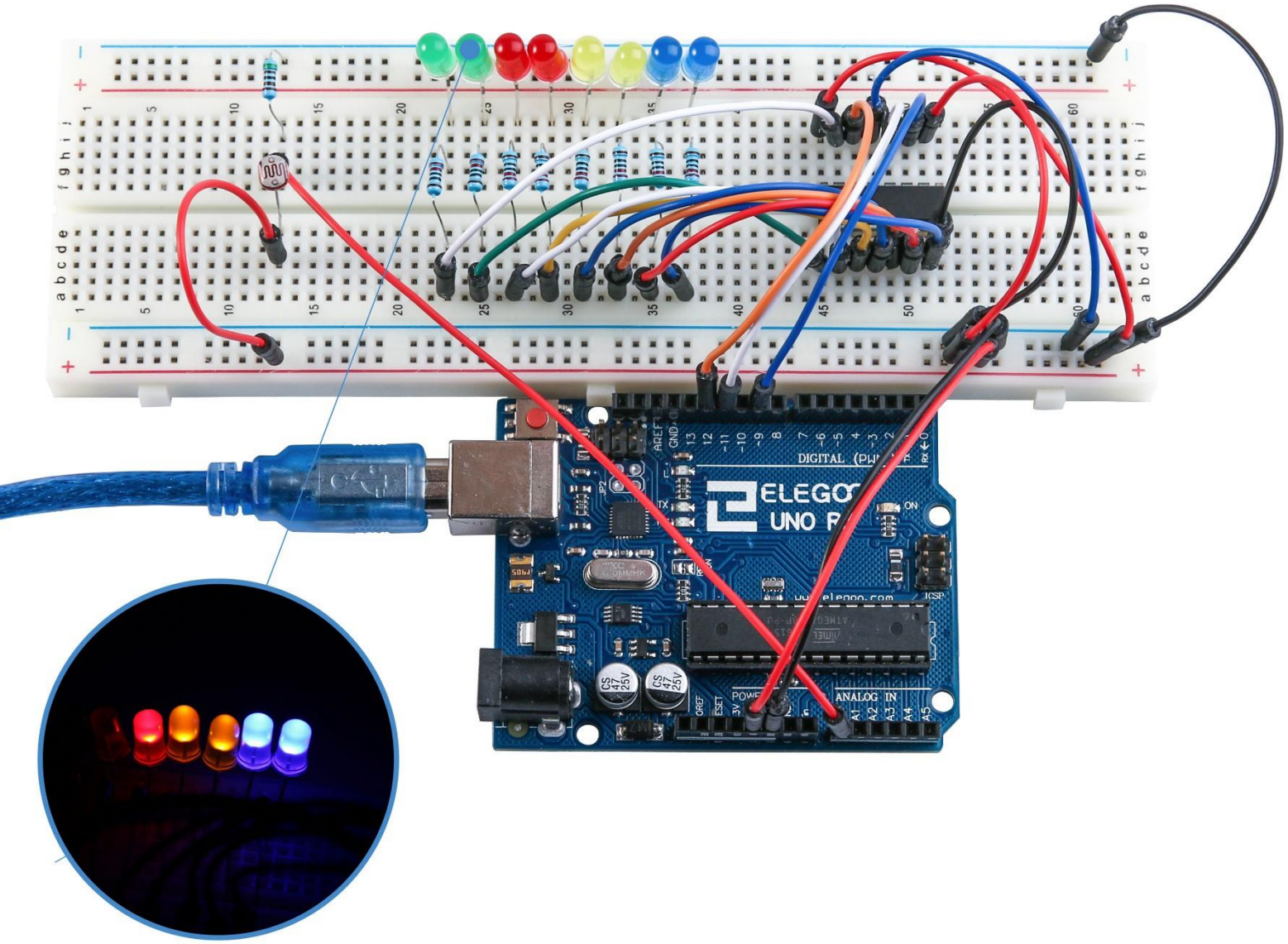
La première chose à remarquer sur le code est qu’il y a la déclaration de la pin qui va permettre de mesure la valeur de la photorésistance : 'lightPin'.

L’autre changement est l’ajout de l’instruction qui calcule le nombre de leds allumées:

```
int numLEDSLit = reading / 57;    // all LEDs lit at 1k
```

Cette fois, nous divisons la valeur brute mesurée par 57 plutôt que 114. Cela permet de créer 9 tranches : de 0 led à 8 leds allumées. Ce facteur est donnée par la valeur de la résistance fixe de 1kΩ, la valeur brute lue sera 1023 / 2 = 511.

Illustration



Leçon 27 74HC595 And Segment Display

But de la leçon

Cette leçon fait suite aux leçons 24,25 et 26. Il s'agit de piloter un afficheur 7 segments.

Matériel nécessaire:

(1) x Elegoo Uno R3

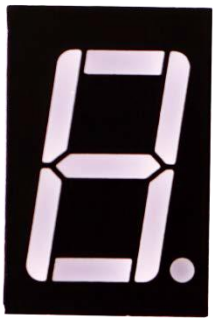
(1) x Planche prototype

(1) x Circuit intégré 74HC595

(1) x 1 Afficheur 7 segments

(8) x Résistance 220 ohms

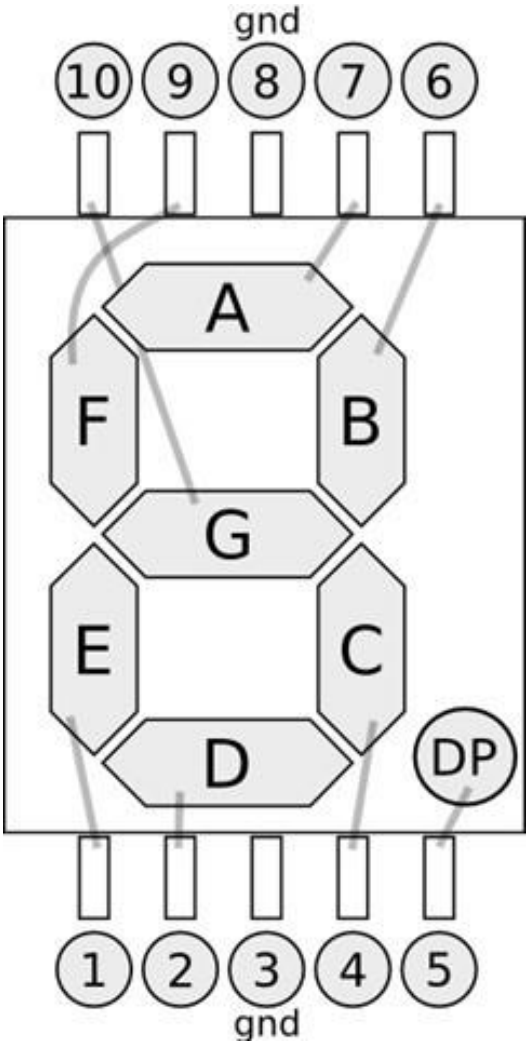
(26) x Câbles mâle-mâle



Présentation du composant

Afficheur 7 segments

Diagramme des pins de l'afficheur:



Le tableau suivant vous permet de connaître les segments à allumer/éteindre en fonction du chiffre à afficher :

Display digital	**Dp**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

Connection

Schéma de câblage

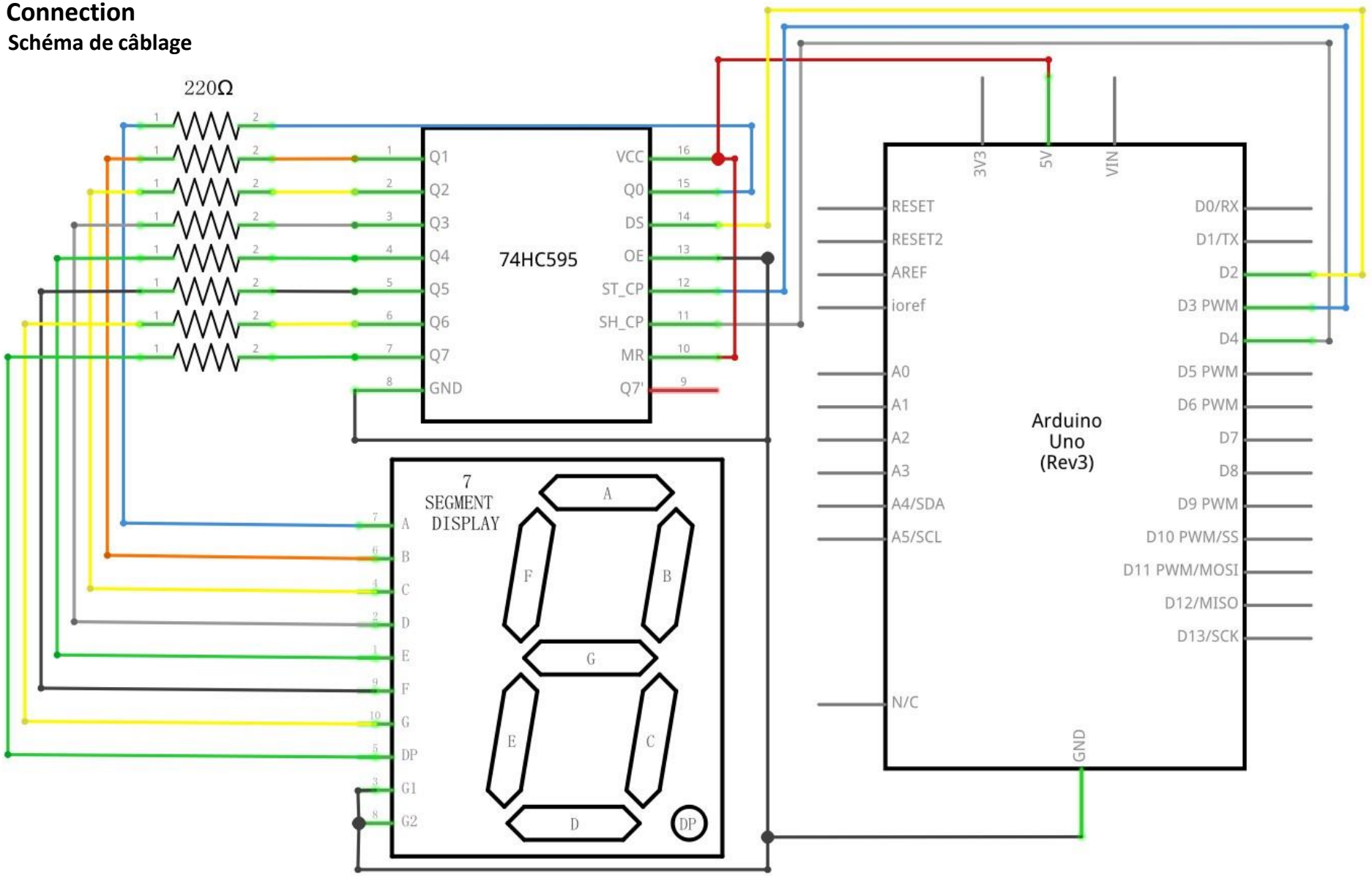
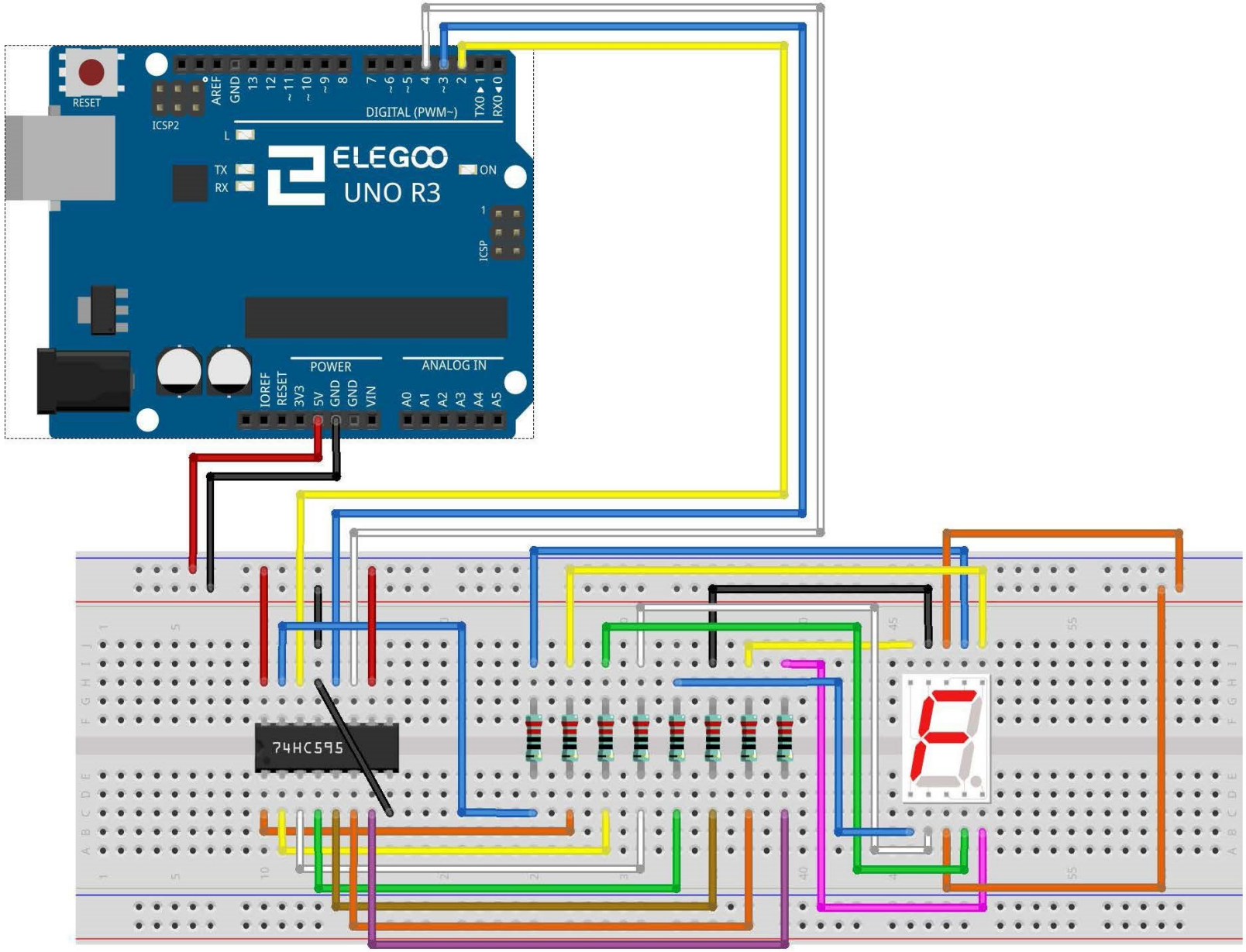


Diagramme de câblage



Le tableau suivant met en concordance les pins du 74HC595 et celles de l'afficheur 7 segments:

74HC595 pin	Seven shows remarkable control pin (stroke)
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

Step 1 : Connectez le 74HC595

VCC (pin 16) et MR (pin 10) connectées à +5V

GND (pin 8) et OE (pin 13) à la masse

Connectez les pins DS, ST_CP et SH_CP:

DS (pin 14) connecté à la carte UNO R3 sur la pin 2 (ligne jaune de l'image)

ST_CP (pin 12, latch pin) connecté à la carte UNO R3 sur la pin 3 (ligne bleue)

SH_CP (pin 11, clock pin) connecté à la carte UNO R3 sur la pin 4 (ligne blanche)

Step 2 : Connectez l'afficheur 7 segments

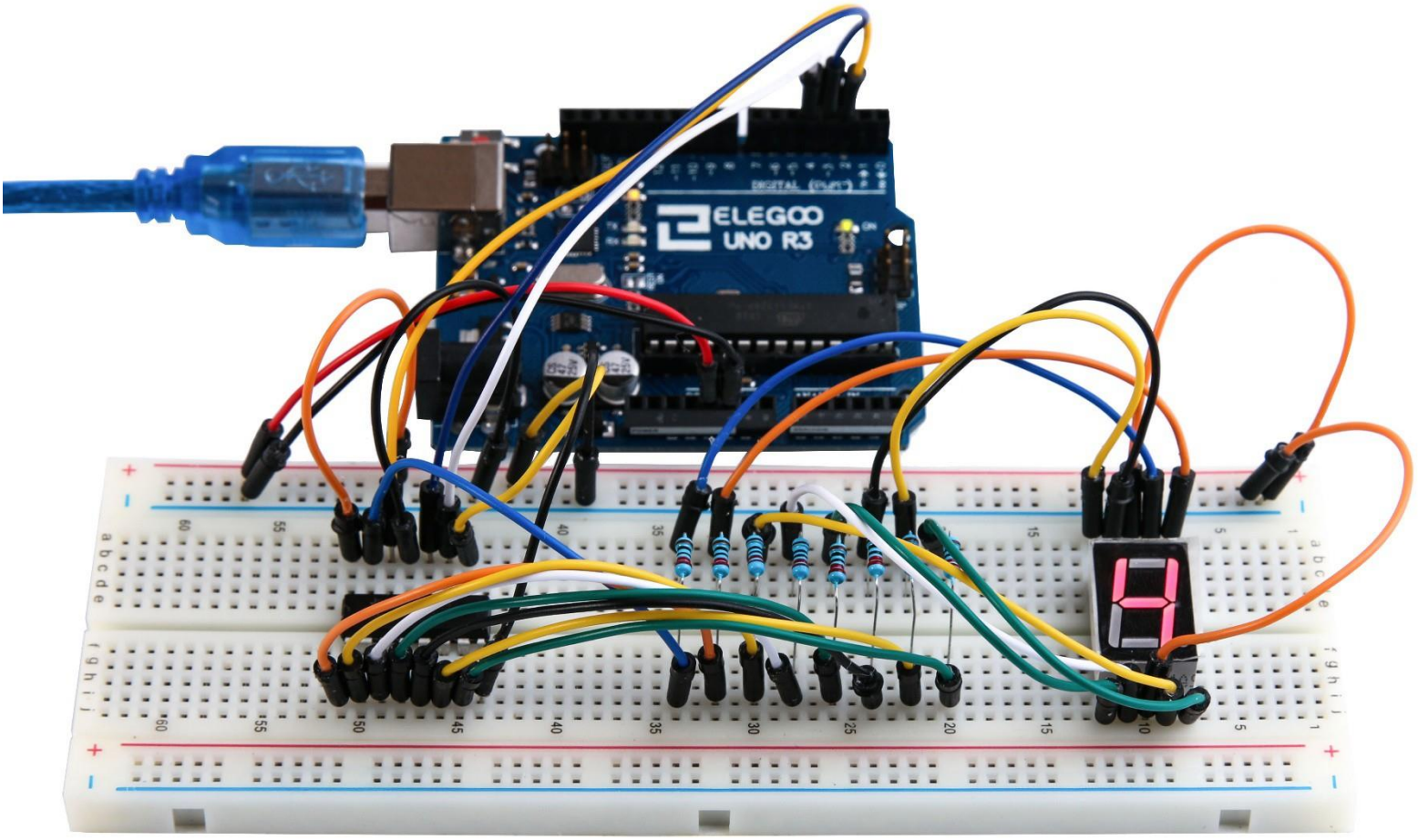
Les pins 3, 8 pin à la carte UNO R3 sur la masse GND (Attention, si l'afficheur est à anode commune le branchement est sur le +5V)

Selon le tableau de la page précédente, connectez le circuit intégré et l'afficheur et placez une résistance 220 ohms en série de chaque segment.

Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 27 74HC595” et Téléversez le code sur la carte UNO R3.

Illustration



Leçon 28 Four Digital Seven Segment Display

But de la leçon

Dans cette leçon, vous apprendrez à utiliser un affichage à 4 chiffres à 7 segments. Lorsque vous utilisez un affichage à 1 segment à 7 segments, notez que s'il s'agit d'une anode commune, la broche commune d'anode se connecte à la source d'alimentation; S'il s'agit d'une cathode commune, la broche cathodique commune se connecte à la GND.

Lorsque vous utilisez un affichage à 4 chiffres à 7 segments, l'anode commune ou la broche cathodique commune est utilisée pour contrôler quel chiffre est affiché. Bien qu'il n'y ait qu'un seul chiffre, le principe de Persistance de Vision vous permet de voir tous les nombres affichés car chaque vitesse de numérisation est si rapide que vous remarquez à peine les intervalles.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x Circuit intégré 74HC595

(1) x Afficheur 4 Digits 7-Segment

(4) x Résistance 220 ohms

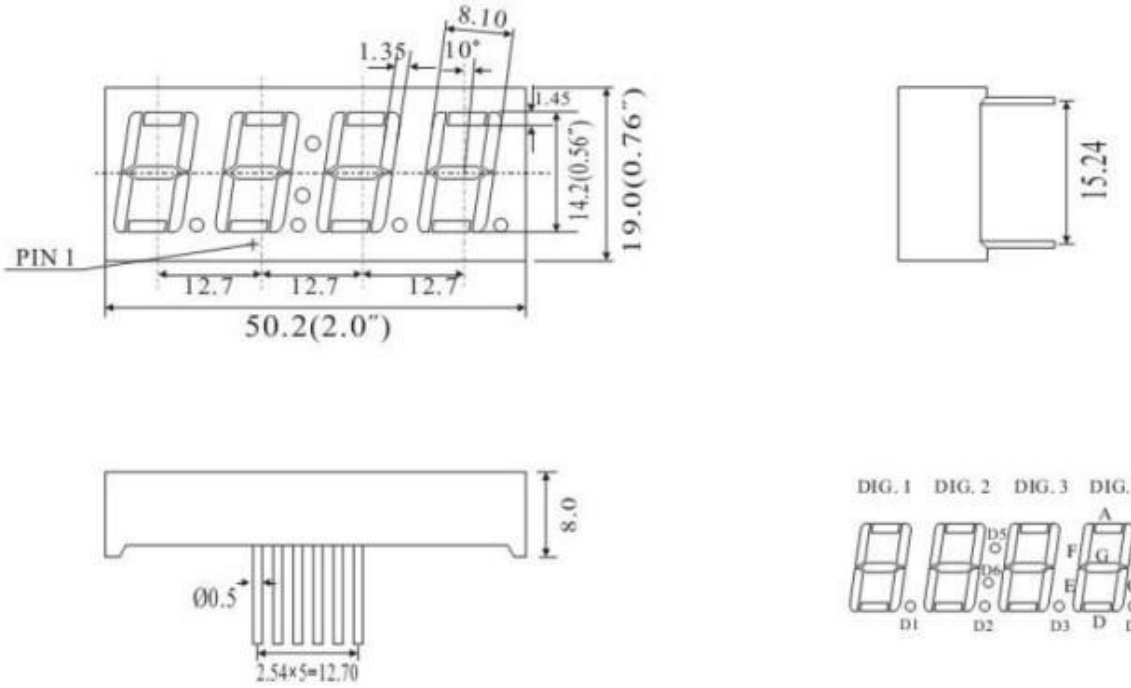
(23) x Câbles mâle-mâle

Présentation du composant

Four Digital Seven segment display

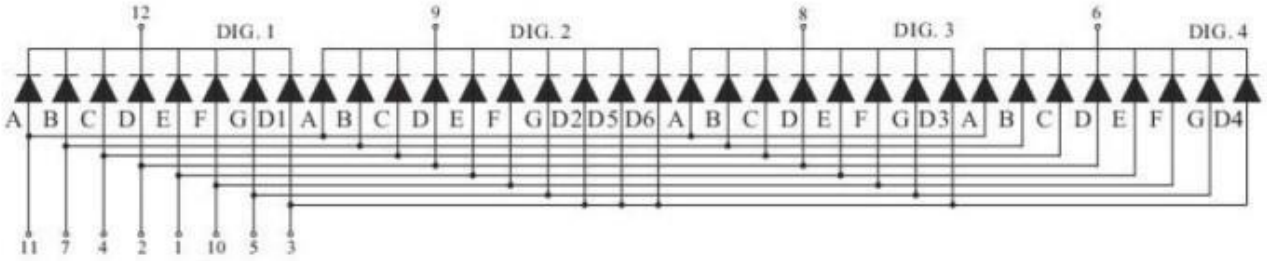
Package Dimensions

CPS05643AB

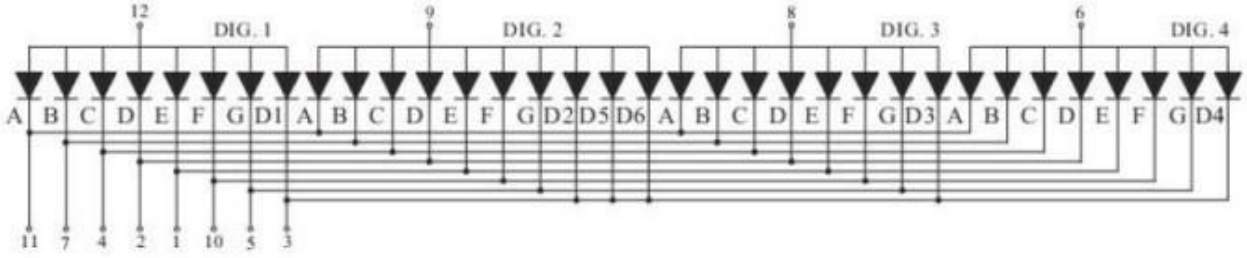


UNIT: MM(INCH) TOLERANCE: ±0.25(0.01")

Internal Circuit Diagram



5643A



5643B

Four Digits Displays Series

Connection

Schéma de câblage

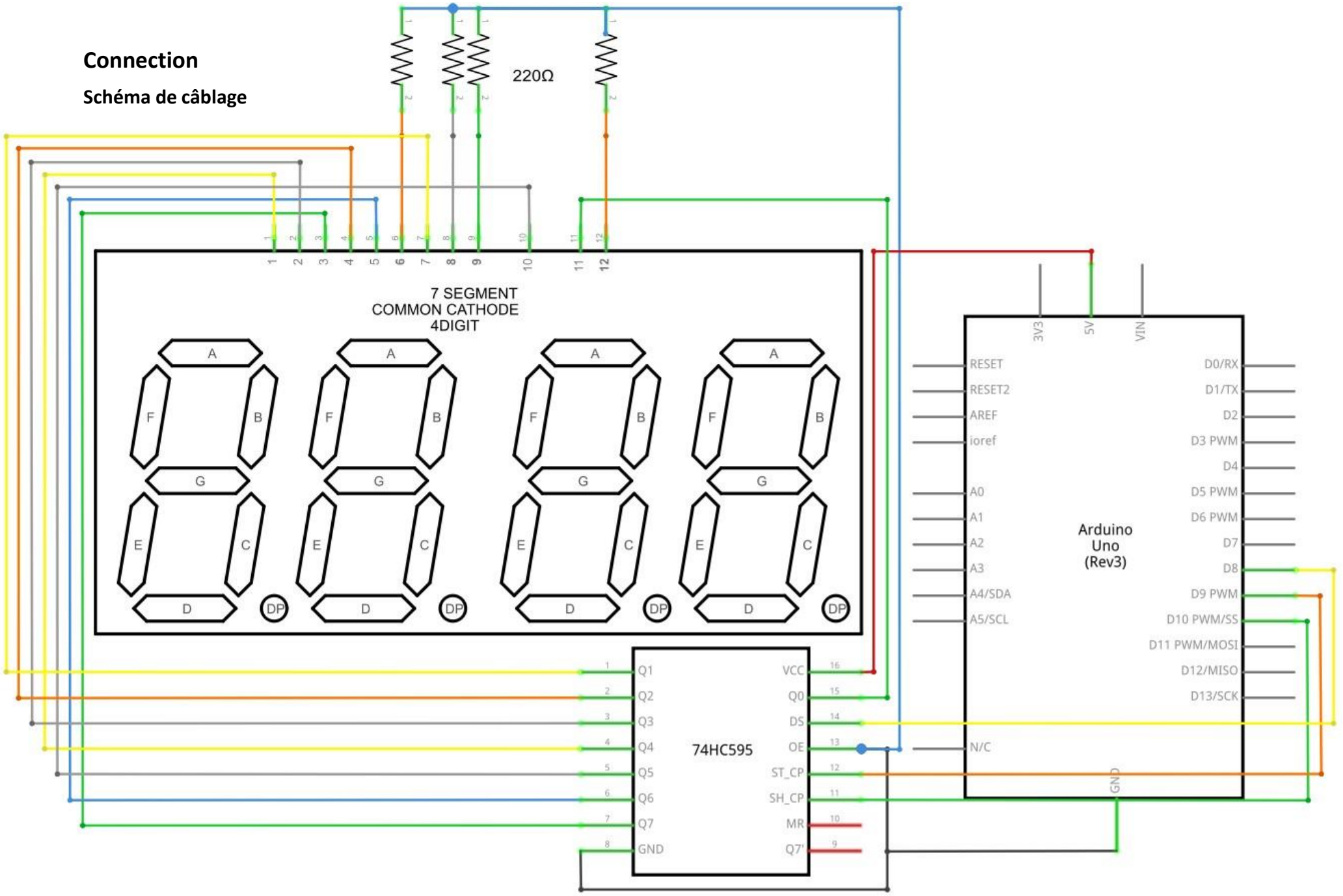
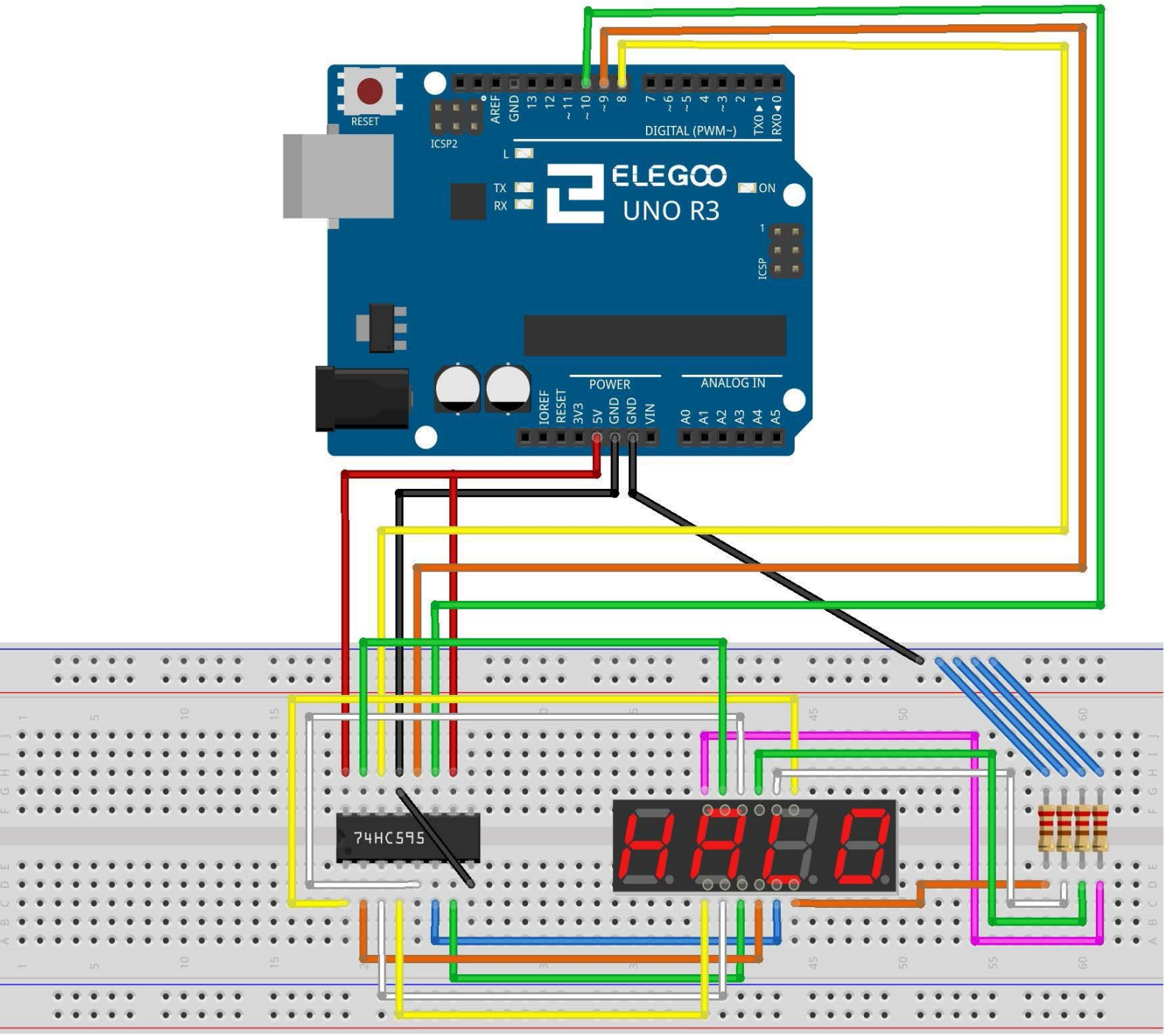


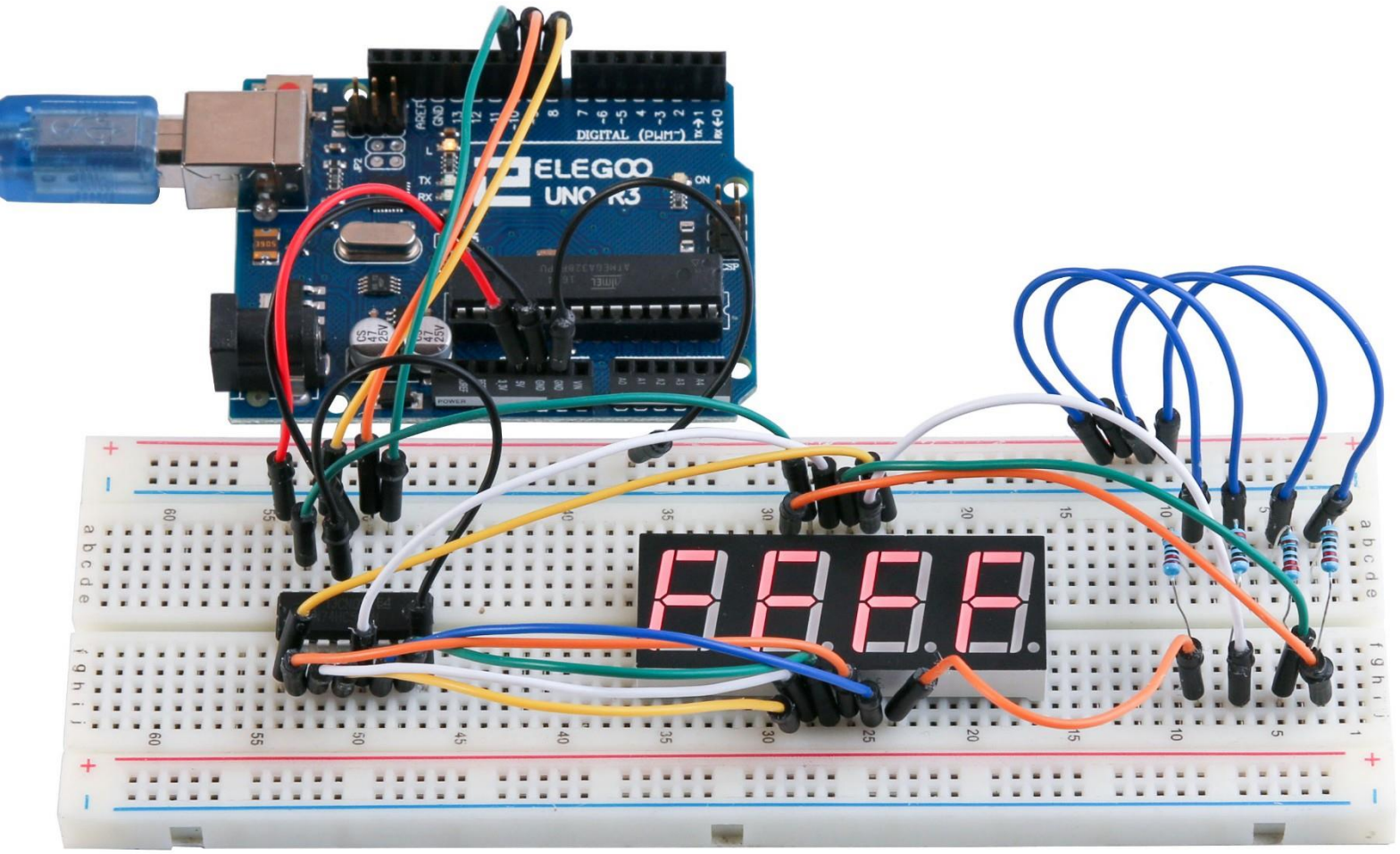
Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 28 Four Digital Seven Segment Display” et Téléversez-le sur la carte UNO R3.

Illustration



Leçon 29 DC Motors

But de la leçon

Dans cette leçon, vous allez apprendre à contrôler un petit moteur à courant continu.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x Circuit intégré L293D

(1) x Moteur 3-6V

(5) x Câbles mâle-mâle

(1) x Alimentation

(1) x Adaptateur 9V1A

Présentation du composant

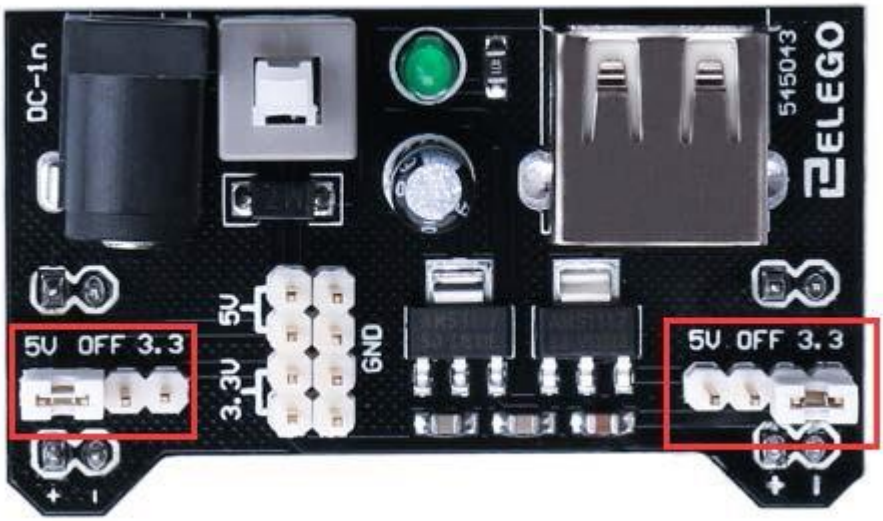
Alimentation

Le moteur nécessite plus de puissance d'alimentation que la carte UNO R3 est en mesure de fournir. C'est pour cela que nous avons besoin d'une alimentation séparée. De plus, si nous connectons directement le moteur à la carte UNO R3, il y a de fortes chances de l'endommager.

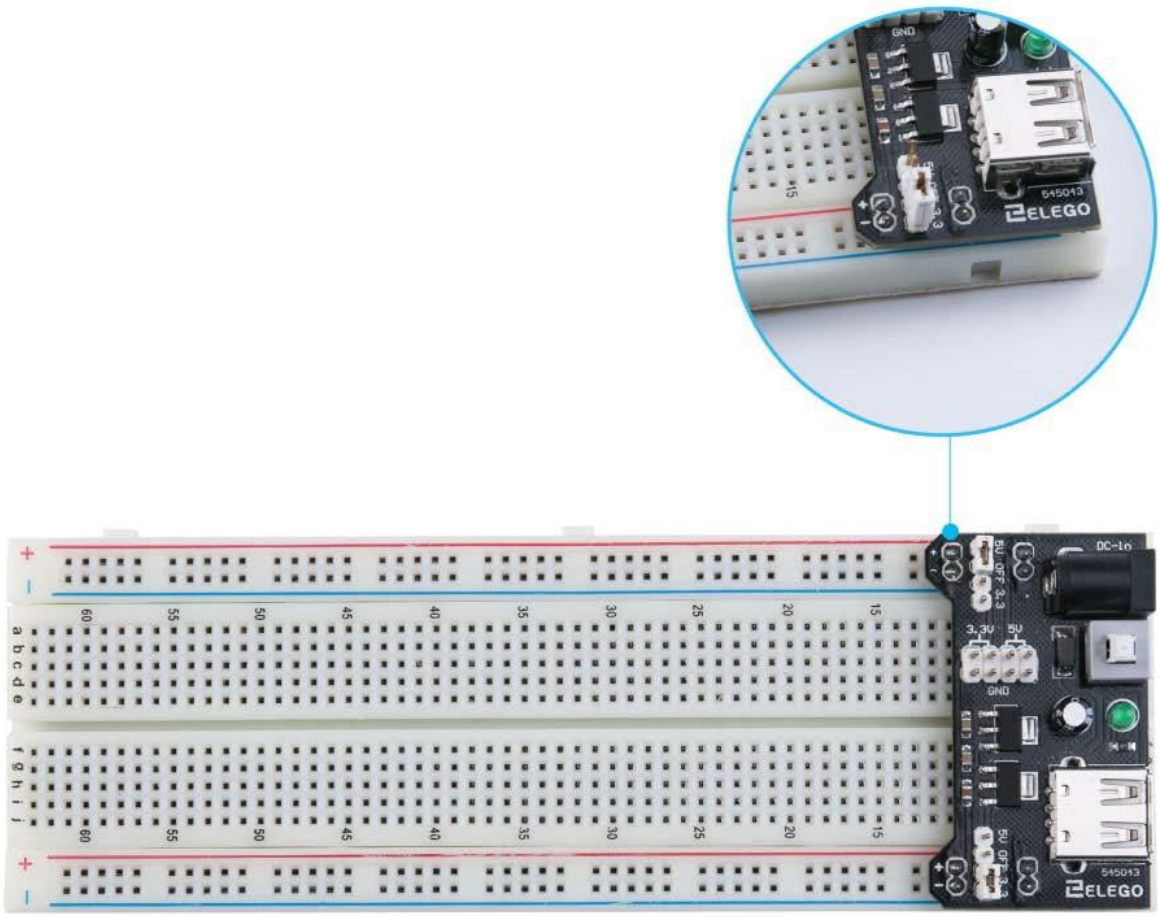
Specifications:

- Bouton On/Off bi-stable
- LED d'indication
- Voltage entrée: 6.5-9v (DC) via connecteur 5.5mm x 2.1mm
- Voltage sortie: 3.3V/5v
- Courant max: 700 mA
- Sortie indépendante. 0v, 3.3v, 5v vers planche prototype
- Size: 2.1 in x 1.4 in
- Connecteur USB pour alimenter des appareils externes

Définir l'alimentation de sortie:



Les jumpers entourés en rouge permettent de définir la puissance du voltage en sortie. A gauche l'alimentation délivrera 5V, à droite 3.3V

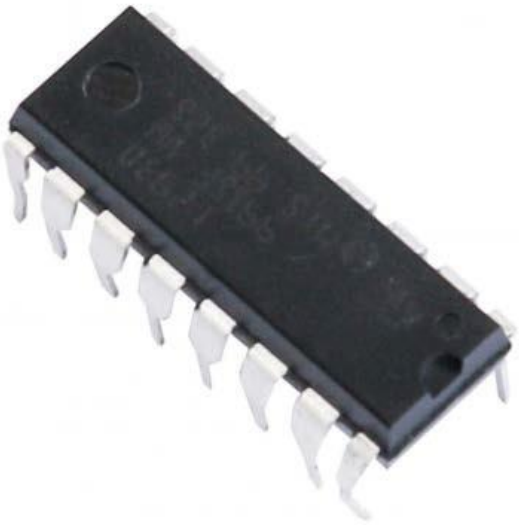


Important:

Faites attention à positionner l'alimentation du bon côté de la planche prototype de telle sorte que le + soit en face du + de la planche, idem pour la masse.

L293D

C'est un composant très courant et utile pour commander des moteurs. Il permet de contrôler 2 moteurs. Comme nous allons en commander un seul, beaucoup de pins seront non connectées.



Spécifications:

- Alimentation de 4.5 à 36V
- Entrées logiques séparées
- Protection contre la surchauffe
- Courant de sortie : 1A (600 mA pour L293D)
- Courant extrême : 2A (1.2 A pour L293D)

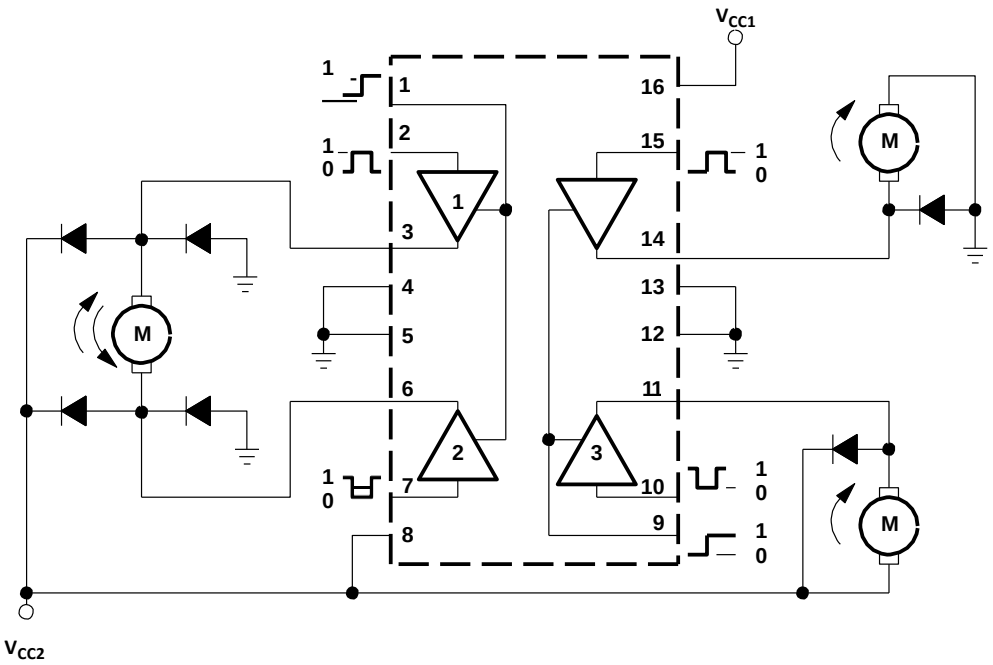
Description:

L293 et L293D sont des quadruples demi-ponts en H. Ils sont conçus pour fournir un courant bi-directionnel de 4.5 à 36V à 1A pour le premier et 0.6A pour le second. Les deux sont conçus pour commander des charges inductives comme les relais, les bobines, les moteurs.

Toutes les entrées sont compatibles TTL.

Les commandes sont activées par paires. Avec 1 et 2 activées par 1,2EN, 3 et 4 activées par 3,4EN. Quand une entrée est à l'état haut, les commandes de sorties associées sont actives. Inversement lorsque l'entrée passe à l'état bas.

Block diagram



Branchements

Moteur 1 PWM : à connecter à une sortie PWM de la carte UNO (ex 5). En jouant avec la valeur de sortie, nous aurons le moteur à l'arrêt (0), ½ de la vitesse max (128), vitesse max (255).

Moteur 1 direction 0/1 et M1 direction 1/0 : connecter ces deux pins à deux sorties digitales. Elles doivent prendre un état logique inverse. En fonction de la valeur, le moteur tournera dans un sens ou dans le sens inverse.

Connection

Schéma de câblage

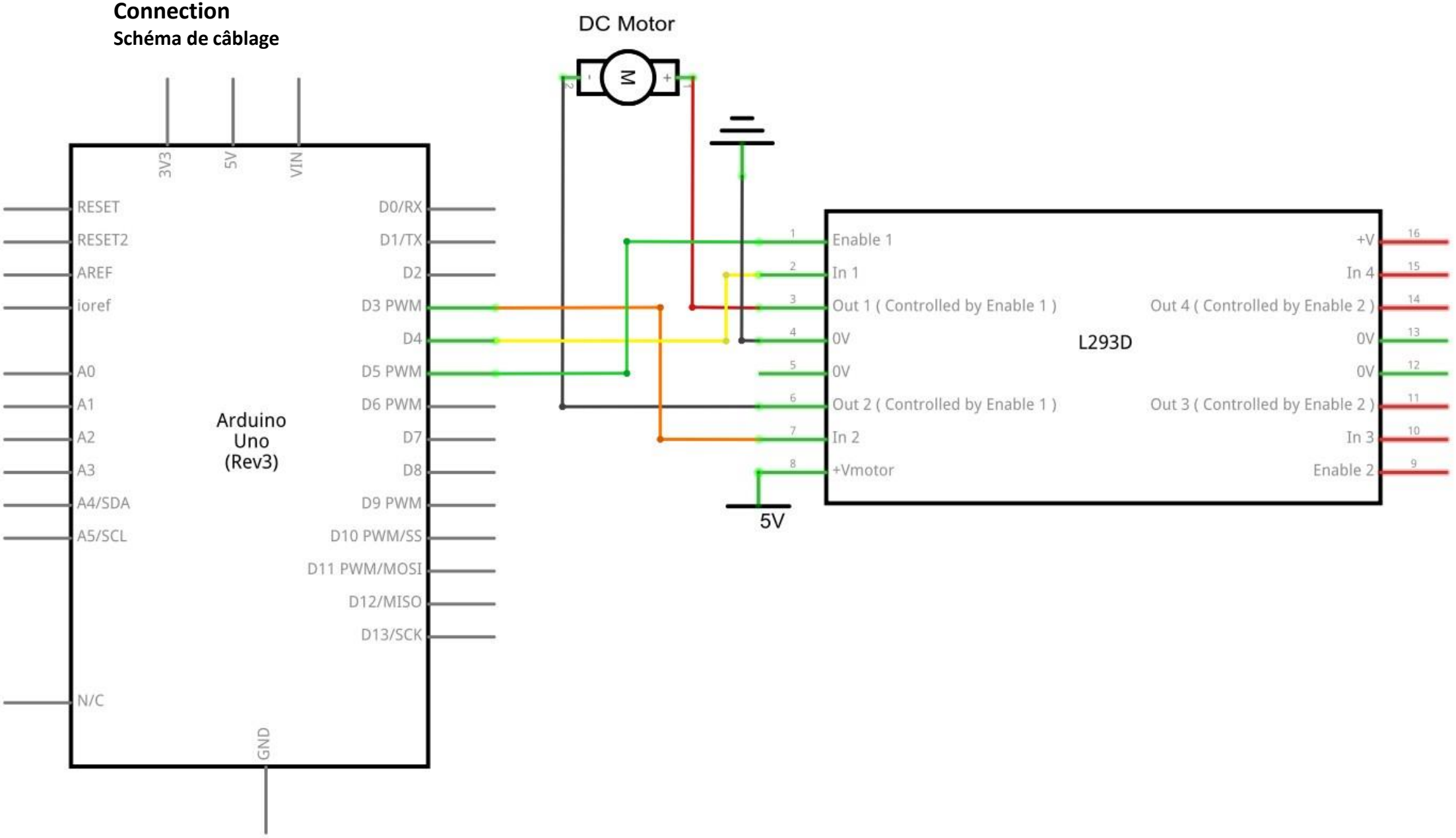
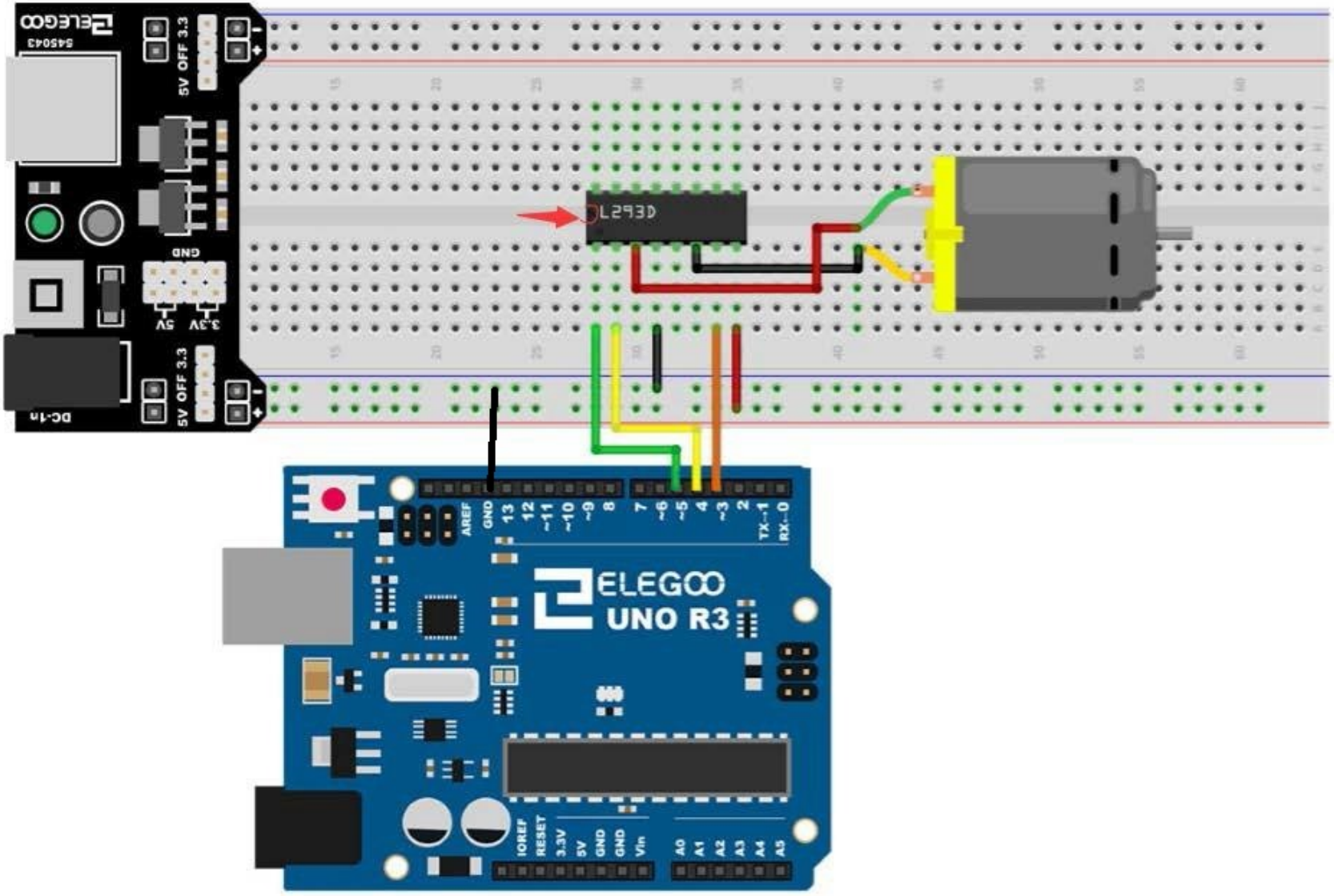
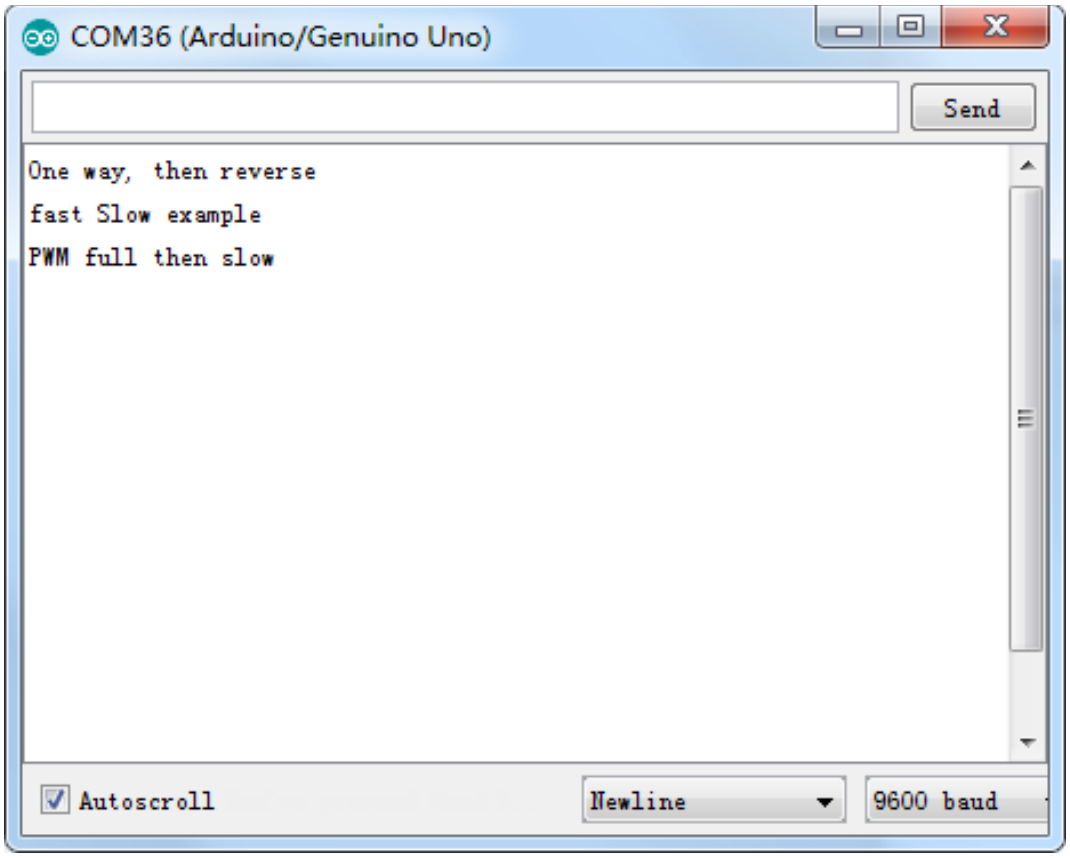


Diagramme de câblage



Rappel : ne jamais connecter un moteur directement à l'alimentation de la carte UNO R3, au risque de l'endommager.

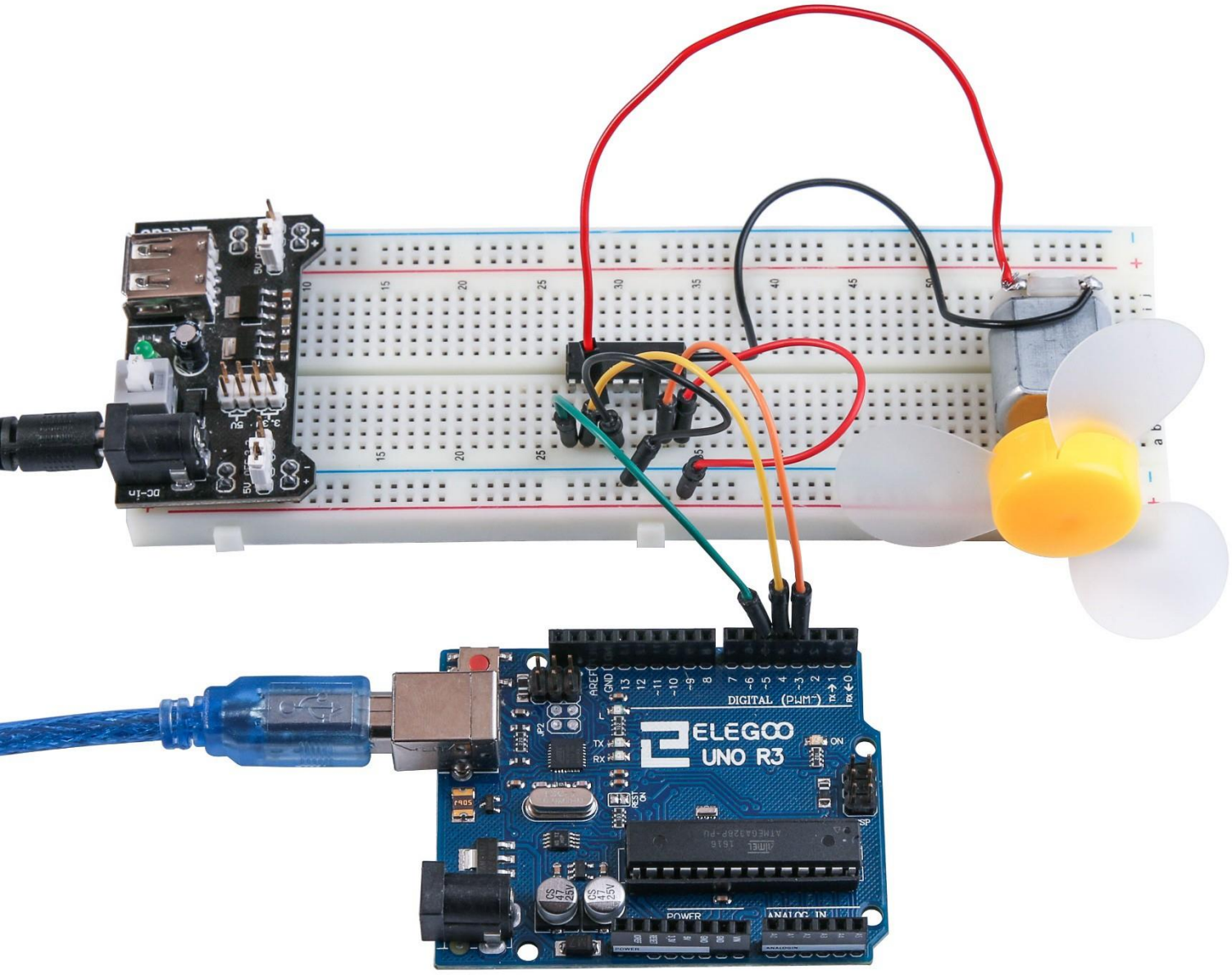


Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon 29 DC Motors" et Téléversez-le sur la carte UNO R3.

Après avoir chargé le programme, allumez l'alimentation, le moteur tourne dans un sens, puis dans l'autre 5x doucement. Puis il tourne avec une vitesse plus élevée dans un sens et dans l'autre. Enfin, il fait une pause de 10s et recommence à zéro.

Illustration



Leçon 30 Relay

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser un relai.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x Moteur

(1) x Circuit intégré L293D

(1) x Relai 5v

(1) x Alimentation

(1) x Adaptateur 9v

(8) x Câbles mâle-mâle

Présentation du composant

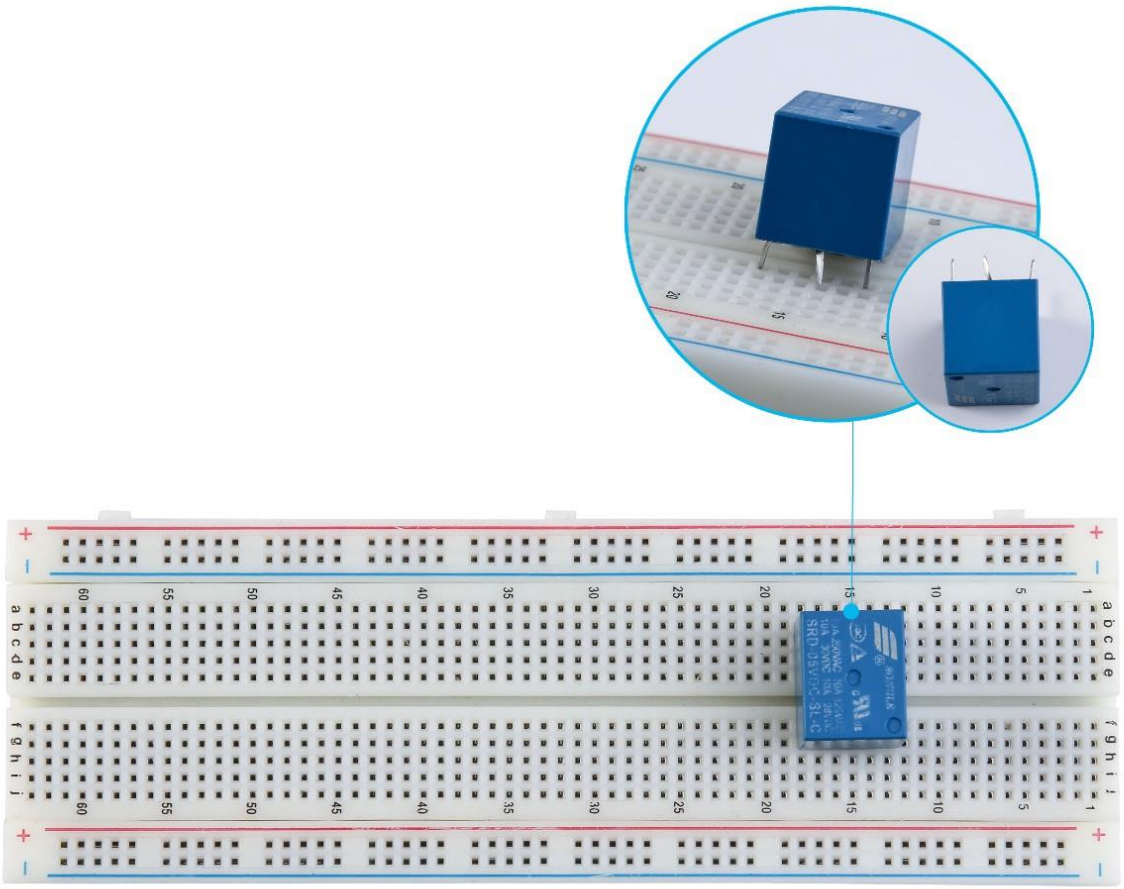
Relai:

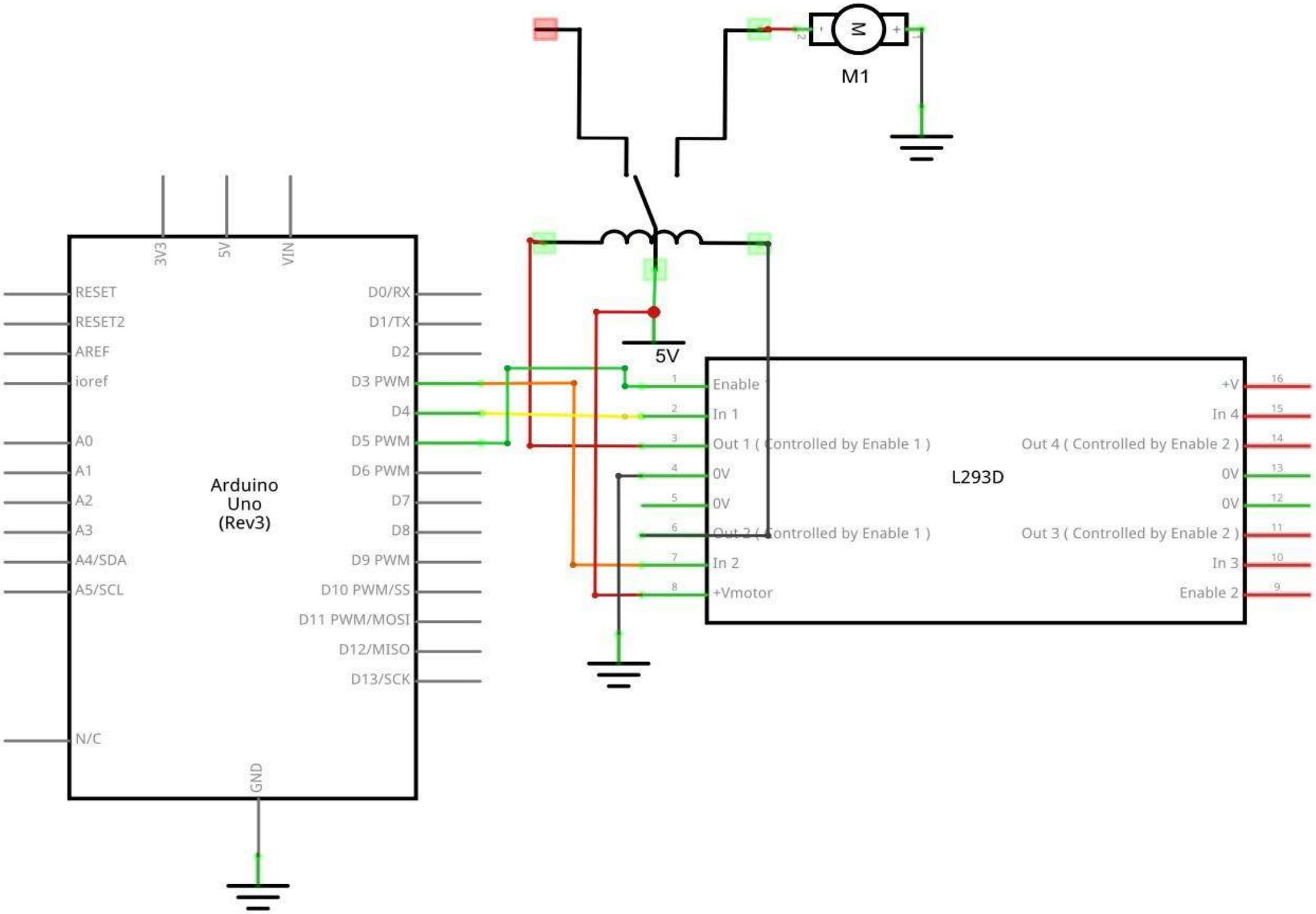
Un relai est un interrupteur commandé électriquement. Beaucoup de relais utilisent l'électromagnétisme pour opérer mécaniquement l'interrupteur, mais il existe d'autres technologies. Les relais sont utilisés notamment lorsque le circuit à commander est isolé électriquement du circuit qui opère le contrôle ou lorsque plusieurs circuit doivent être commandés par un signal unique.

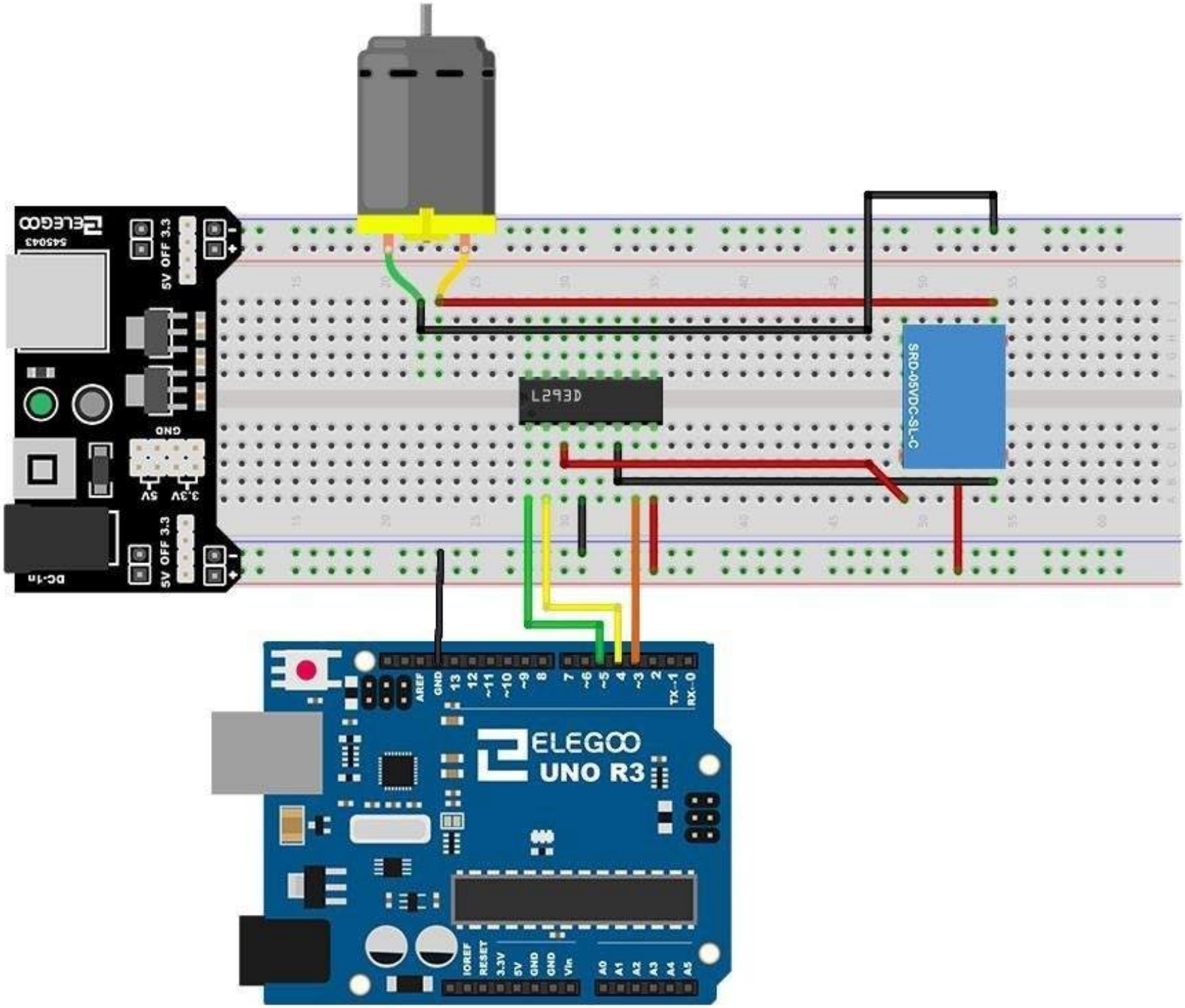
Un type de relai qui peut manipuler des circuits à « haute » tension sont appelés les contacteurs. Ils sont faits pour supporter les surcharges électriques. Dans l'électronique moderne, ces fonctions sont produites par des instruments numériques appelés « relais de protection ».

Le schéma ci-dessous montre comment on câble un relai avec une carte UNO R3.

Faites bien attention lors de la mise en place du relai.




M1
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
5V
Enable 1
In 1
Out 1 (Controlled by Enable 1)
0V
0V
Out 2 (Controlled by Enable 1)
In 2
+Vmotor
L293D
+V
In 4
Out 4 (Controlled by Enable 2)
0V
0V
Out 3 (Controlled by Enable 2)
In 3
Enable 2

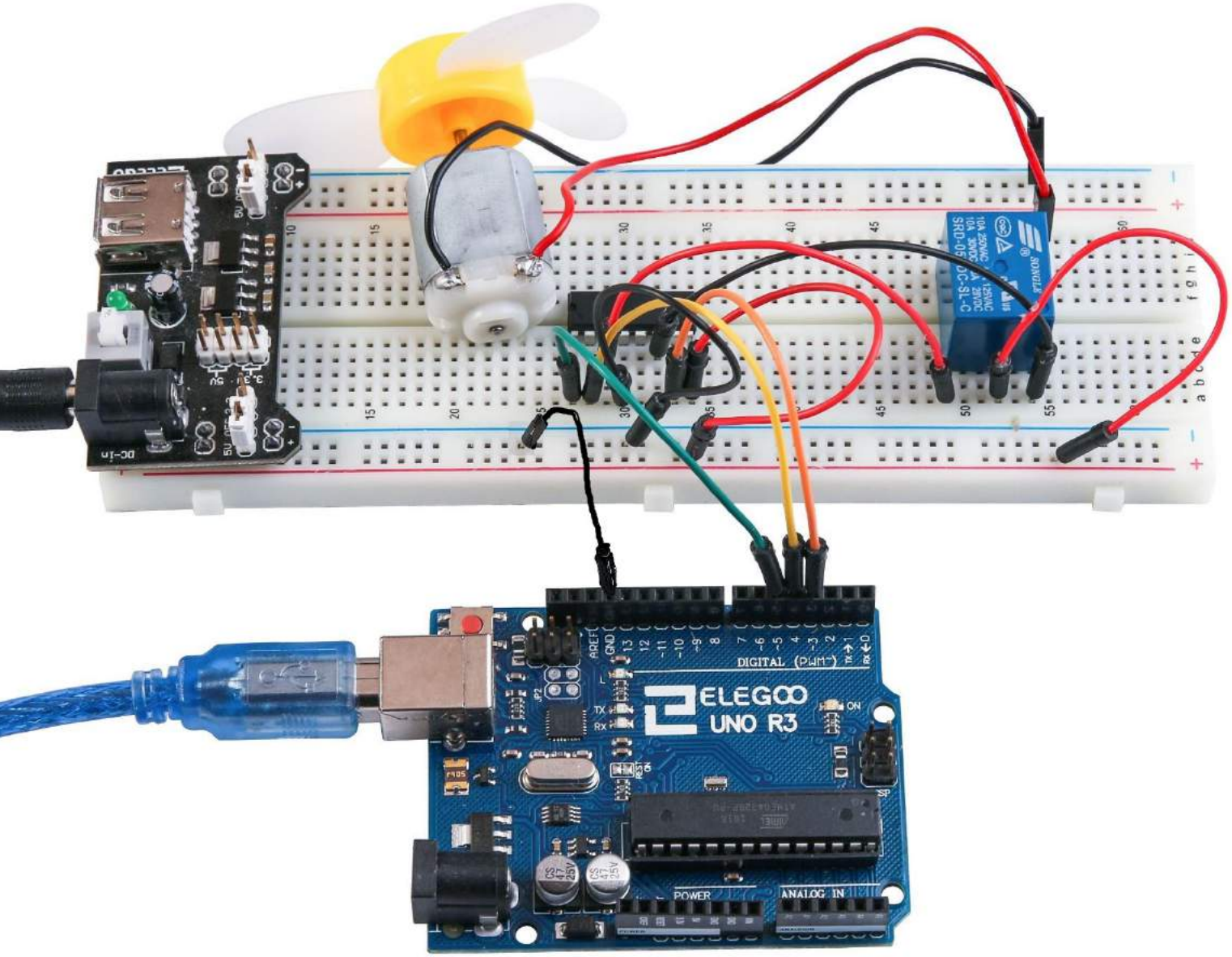

L293D
SRD-05VDC-SL-C
ELEGOO
UNO R3

Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 30 Relay” et téléversez-le sur la carte UNO R3.

Après avoir mis sous tension l’alimentation, vous pourrez entendre le bruit du relai, le moteur se met en route et après un moment, il se coupe.

Illustration



Leçon 31 Stepper Motor

But de la leçon

Dans cette leçon vous allez apprendre à utiliser un moteur pas à pas.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x Driver moteur ULN2003

(1) x Moteur pas à pas

(1) x Adaptateur 9V

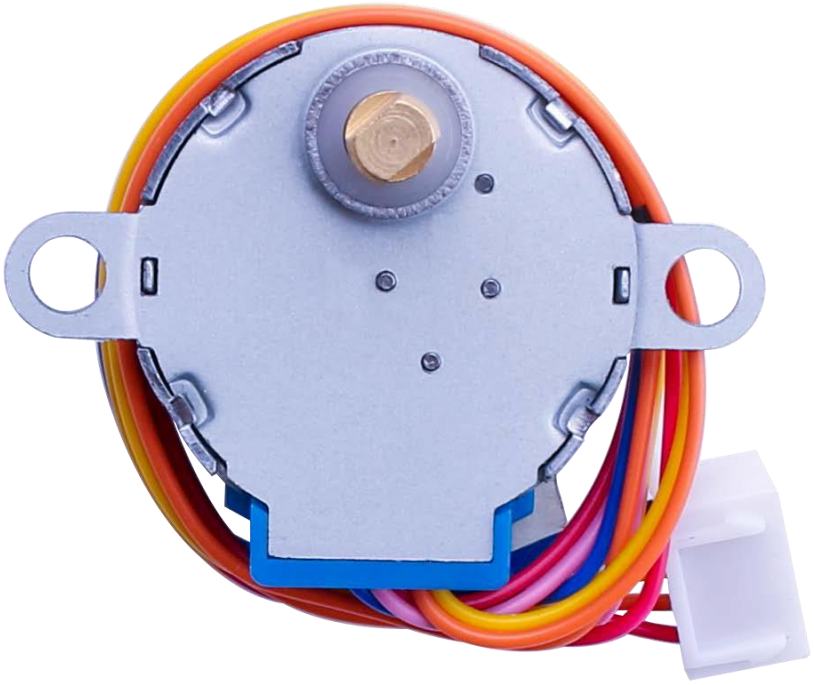
(1) x Alimentation

(6) x Câbles Mâle-Femelle

(1) x Câbles Mâle-Mâle

Présentation du composant

Moteur pas à pas

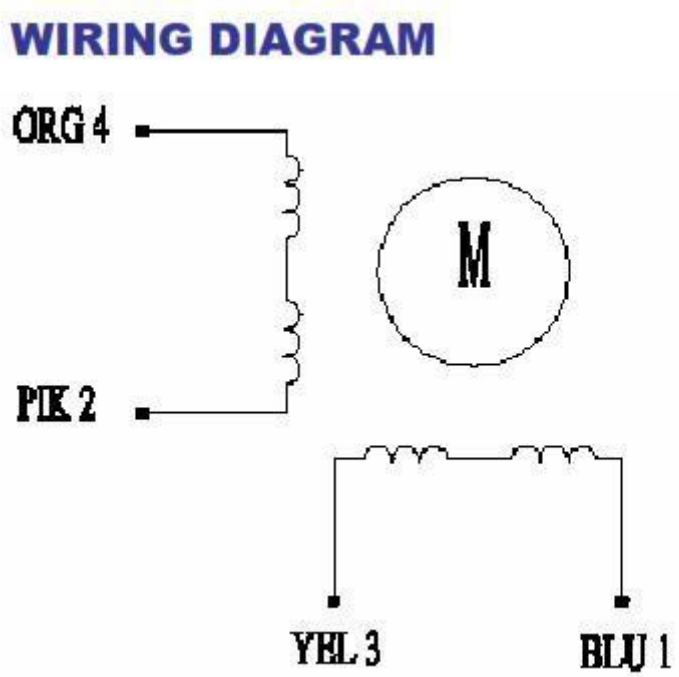


Un moteur pas à pas est une machine électromagnétique qui converti des impulsions électriques en mouvement précis. L'arbre du moteur bouge par petites incrémentations quand la pulsation électrique est appliquée. La rotation du moteur a un lien direct avec l'impulsion électrique. Il en est de même pour la vitesse. L'avantage du moteur pas à pas est de pouvoir exécuter des mouvements très précis.

Stepper motor 28BYJ-48 Parameters

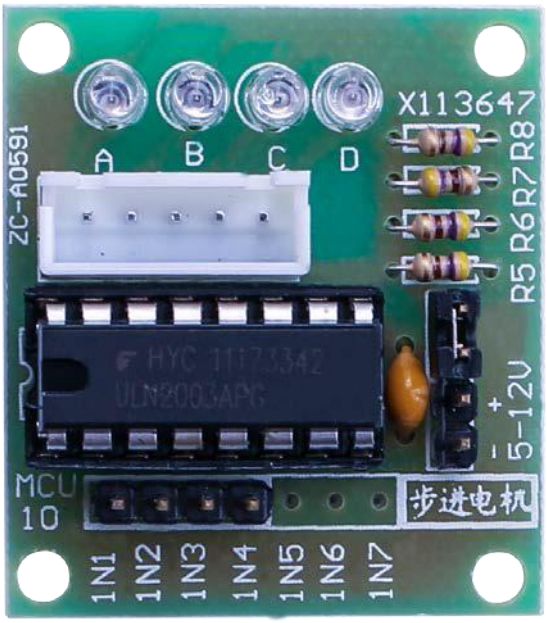
- Model: 28BYJ-48
- Voltage: 5VDC
- Phases: 4
- Ratio de variation de vitesse: 1/64
- Angle: 5.625° /64
- Fréquence: 100Hz
- Résistance au courant continu: 50Ω±7%(25℃)
- Fréquence In-traction: > 600Hz
- Fréquence Out-traction: > 1000Hz
- Couple >34.3mN.m(120Hz)
- Couple de friction: 600-1200 gf.cm
- Résistance induite >10MΩ(500V)
- Courant induit： 600VAC/1mA/1s
- Grade induit： A
- Température <40K(120Hz)
- Bruit <35dB(120Hz,No load,10cm)

Interfacing circuits



Le moteur pas à pas bipolaire a généralement quatre fils qui en découlent. Contrairement aux steppers unipolaires, les steppers bipolaires n'ont pas de connexion centrale commune. Ils ont plutôt deux ensembles indépendants de bobines. Vous pouvez les distinguer des steppes unipolaires en mesurant la résistance entre les fils. Vous devriez trouver deux paires de fils avec une résistance égale. Si vous avez les fils de votre compteur connectés à deux fils qui ne sont pas connectés (c'est-à-dire non attachés à la même bobine), vous devriez voir une résistance infinie (ou aucune continuité).

Carte ULN2003



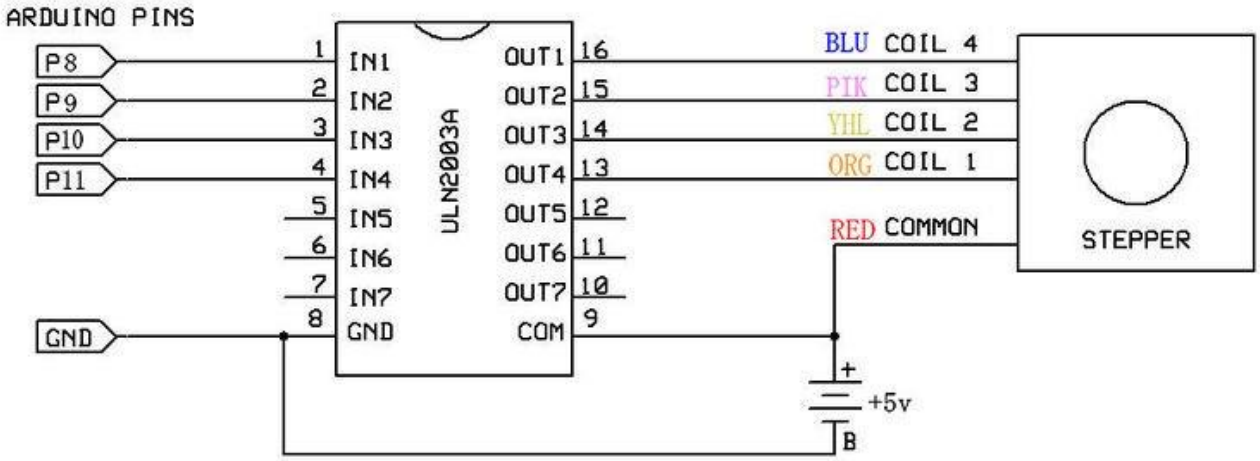
Description

- Dimensions: 42mmx30mm
- Puce ULN2003 500mA
- A. B. C. D LED indication des phases de fonctionnement du moteur.

La méthode la plus simple pour interfacer un moteur pas à pas sur une carte UNO est d'utiliser une carte de type ULN2003A .

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Schéma de câblage



Connection

Schéma de câblage

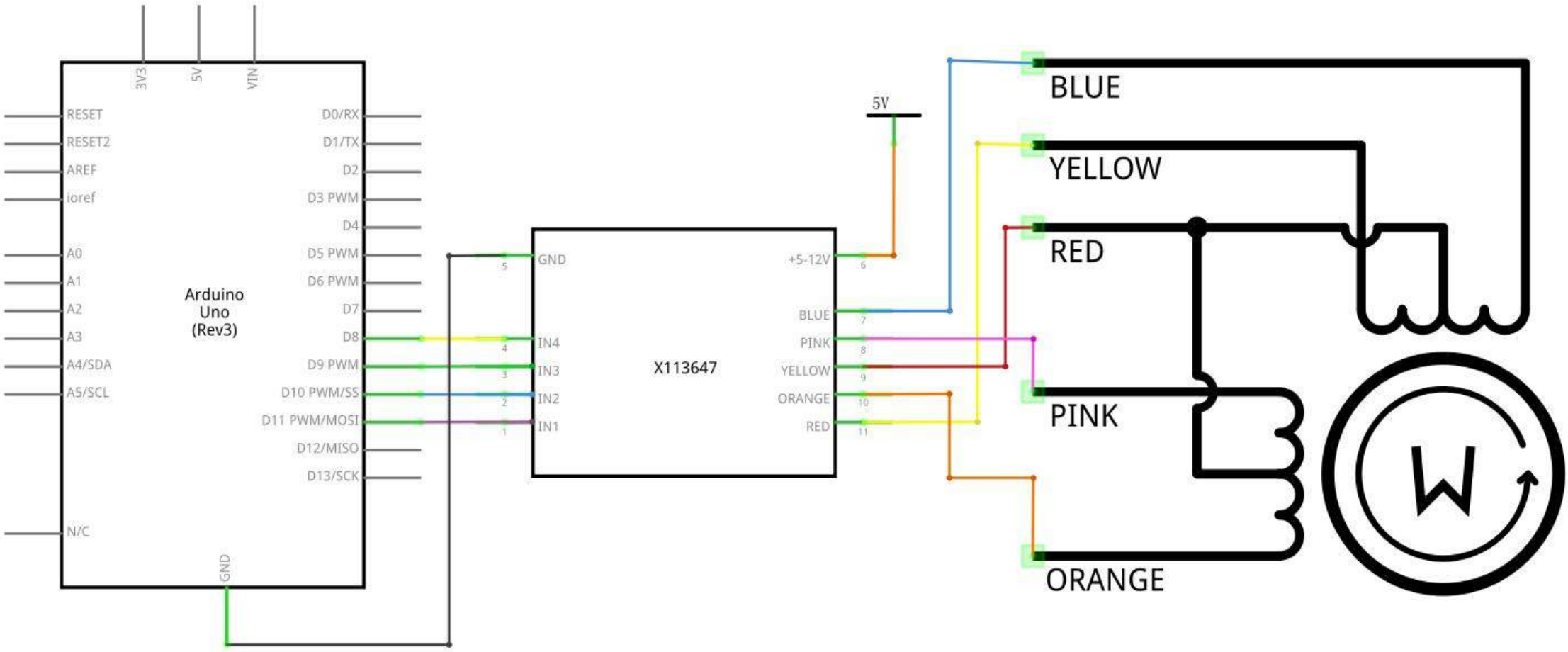
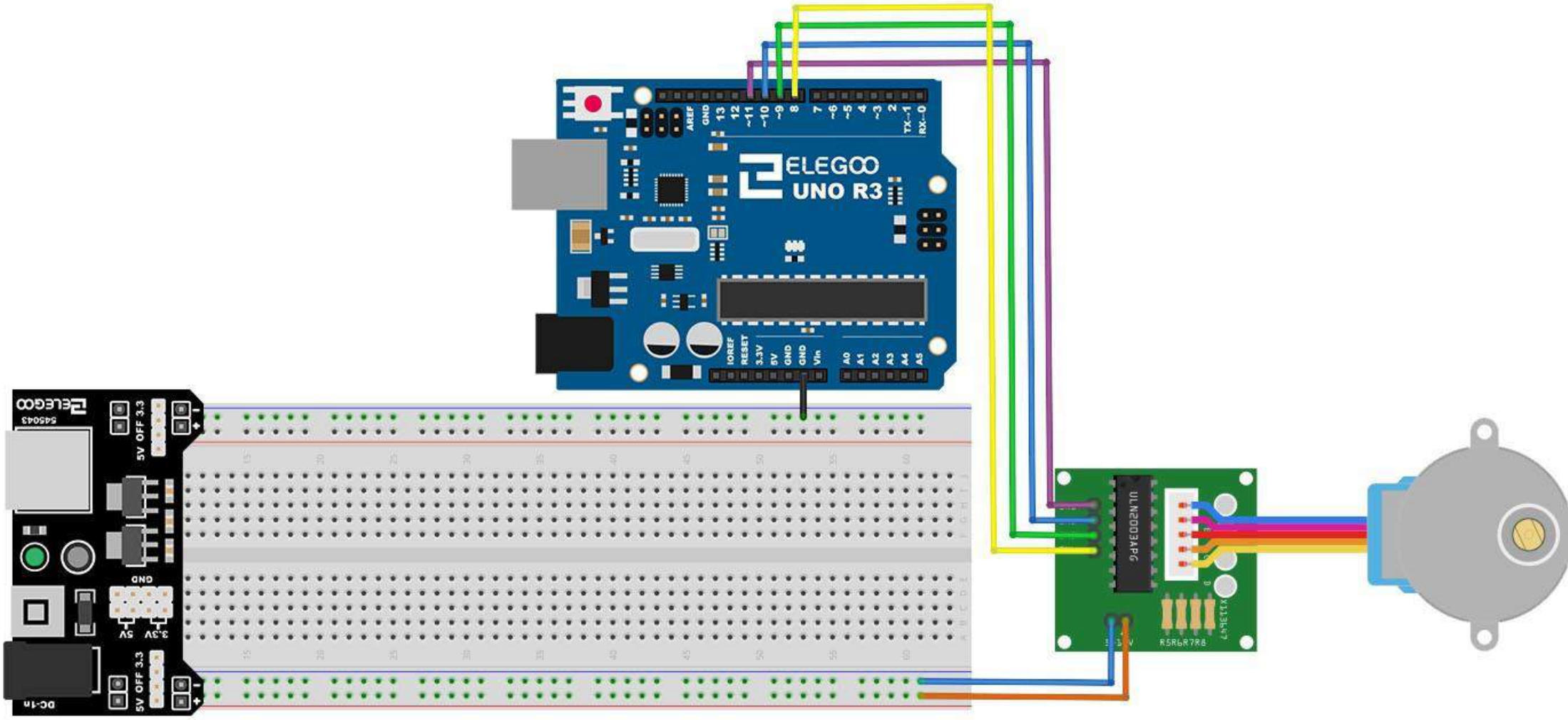


Diagramme de câblage

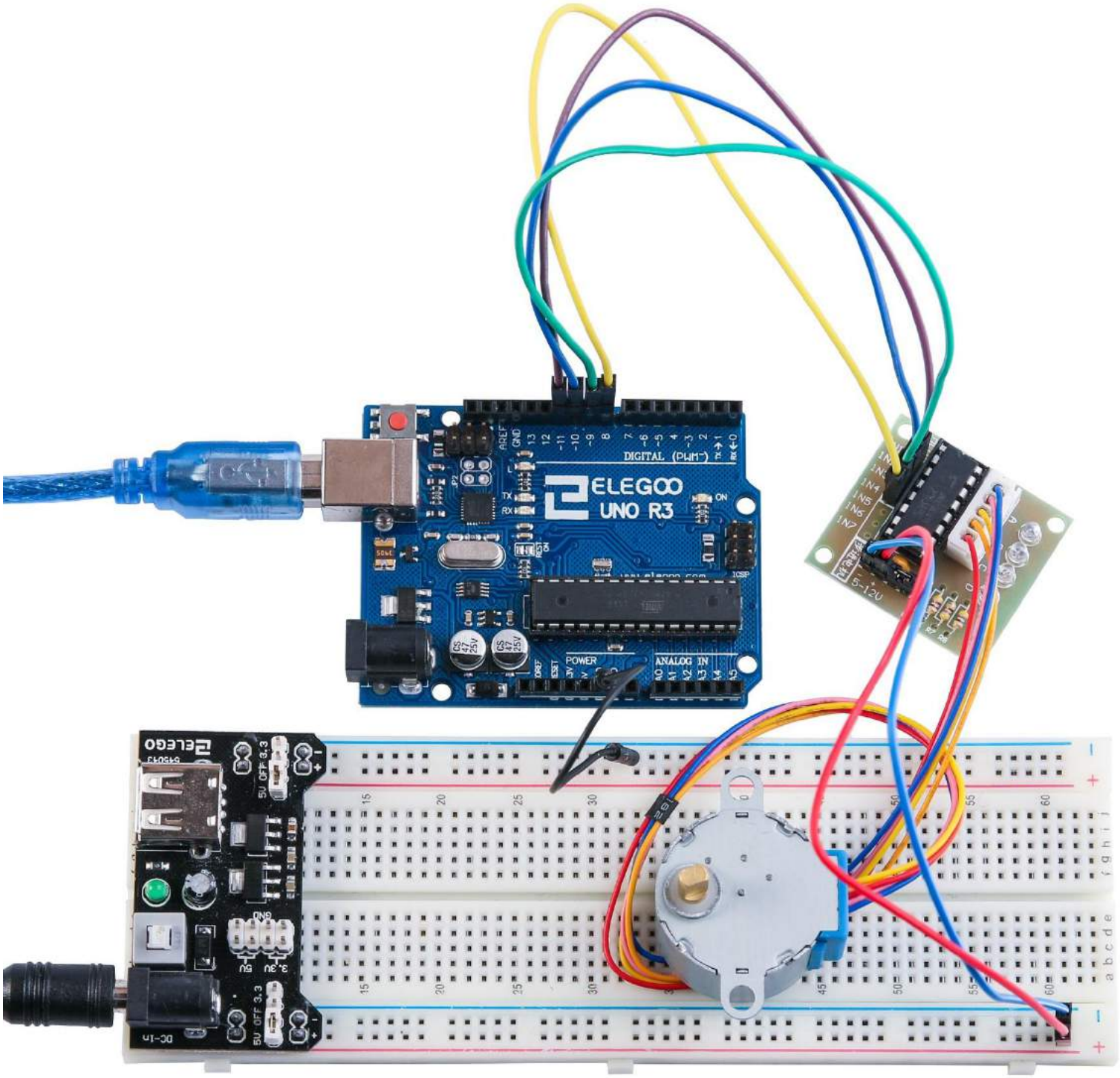


Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 31 Stepper” et Téléversez-le sur la carte UNO R3.

Veillez à bien avoir installé la bibliothèque < Stepper > avant toute chose. En cas de difficulté, reportez-vous à la leçon 1.

Illustration



Leçon 32 Controlling Stepper Motor With Remote

But de la leçon

Partant de la leçon précédente, vous allez apprendre dans cette leçon comment commander le moteur avec une télécommande.

Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x Planche prototype
(1) x Récepteur infrarouge
(1) x Télécommande infrarouge
(1) x Carte ULN2003
(1) x Moteur pas à pas
(1) x Alimentation
(1) x Adaptateur 9V
(9) x Câbles Mâle-Femelle
(1) x Câbles Mâle-Mâle

Connection

Schéma de câblage

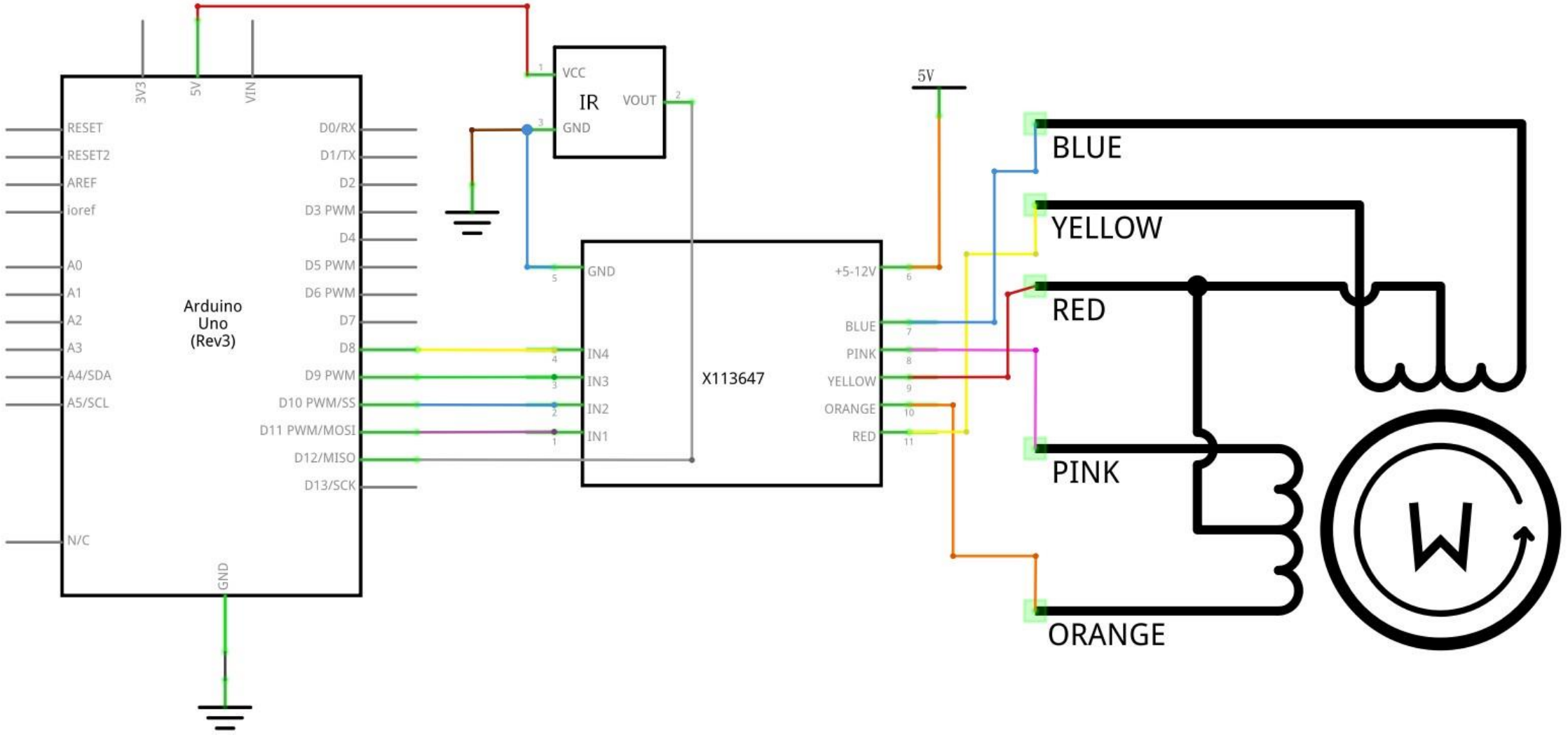
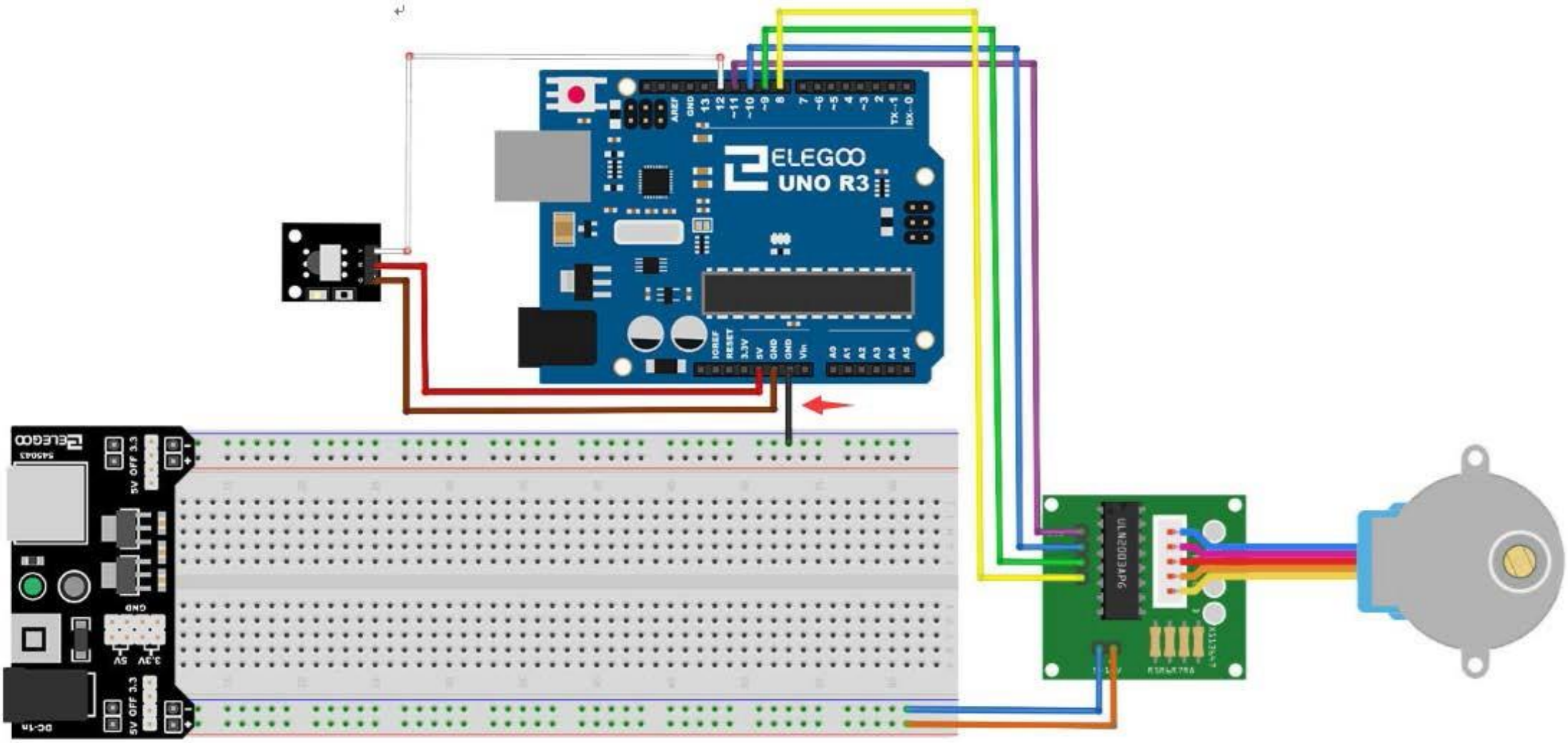


Diagramme de câblage



4 pins sont nécessaires pour le moteur, une pour le récepteur infrarouge.

Les pins 8 à 11 sont pour le moteur, la pin 12 est pour le récepteur.

Attention, le +5V et la masse pour le moteur sont ceux de l'alimentation externe.

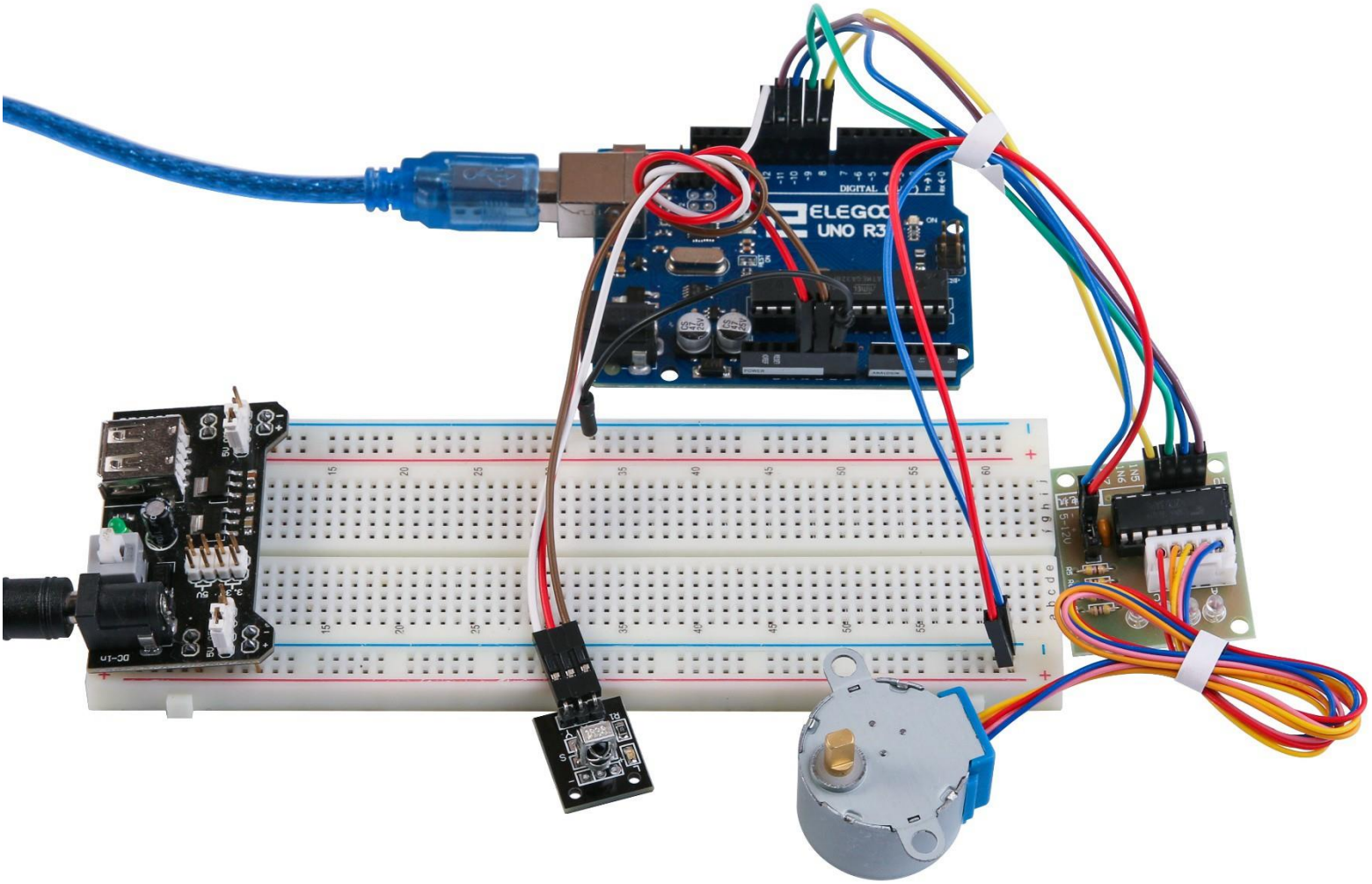
Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon 32 Controlling Stepper Motor With Remote" et téléversez-le sur la carte Uno R3.

Avant toute chose, veillez à bien avoir installé la bibliothèque < IRremote > et la bibliothèque < Stepper >. Reportez-vous à la leçon 1 en cas de difficulté.

VOL+ de la télécommande fera tourner le moteur dans un sens, et VOL- en sens inverse.

Illustration



Leçon 33 Controlling Stepper Motor With Rotary Encoder

But de la leçon

Cette leçon fait suite au deux précédentes.

Vous allez apprendre comment réaliser une rotation du moteur grâce à un encodeur rotatif.

Le matériel nécessaire est le même que celui de la leçon 32. On y ajoute simplement l'encodeur.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x Module encodeur

(1) x Carte ULN2003

(1) x Moteur pas à pas

(1) x Alimentation

(1) x Adaptateur 9V

(9) x Câbles Mâle-Femelle

(1) x Câbles Mâle-Mâle

Présentation du composant

Encodeur rotatif

Il s'agit d'un composant électromécanique qui converti un mouvement de rotation en signal numérique.

Il y en a deux types : absolu ou à incrément. La sortie d'un encodeur absolu indique la position actuelle de l'arbre, faisant la conversion en angle. La sortie d'un encodeur à incrément fourni une information concernant le mouvement de l'arbre.

Les encodeurs rotatifs sont utilisés dans des applications diverses qui nécessite une information de mouvement très précise.

Connection

Schéma de câblage

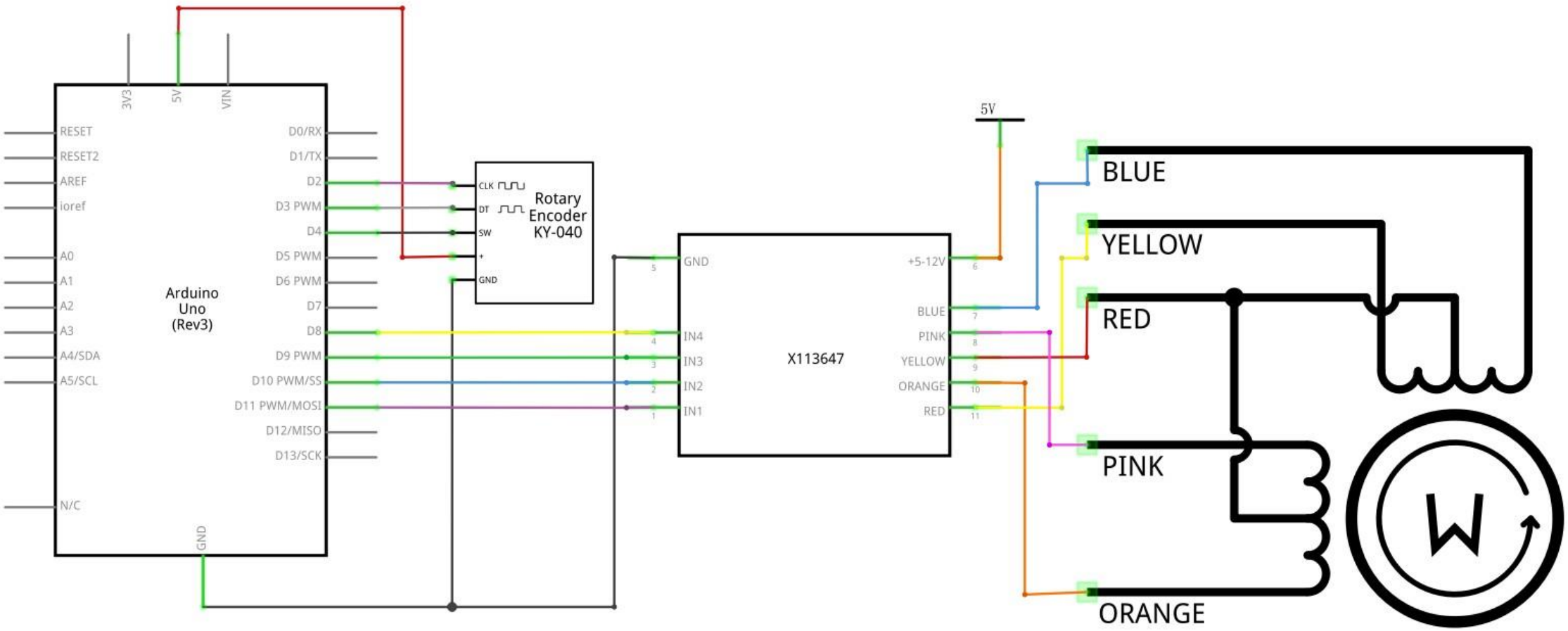
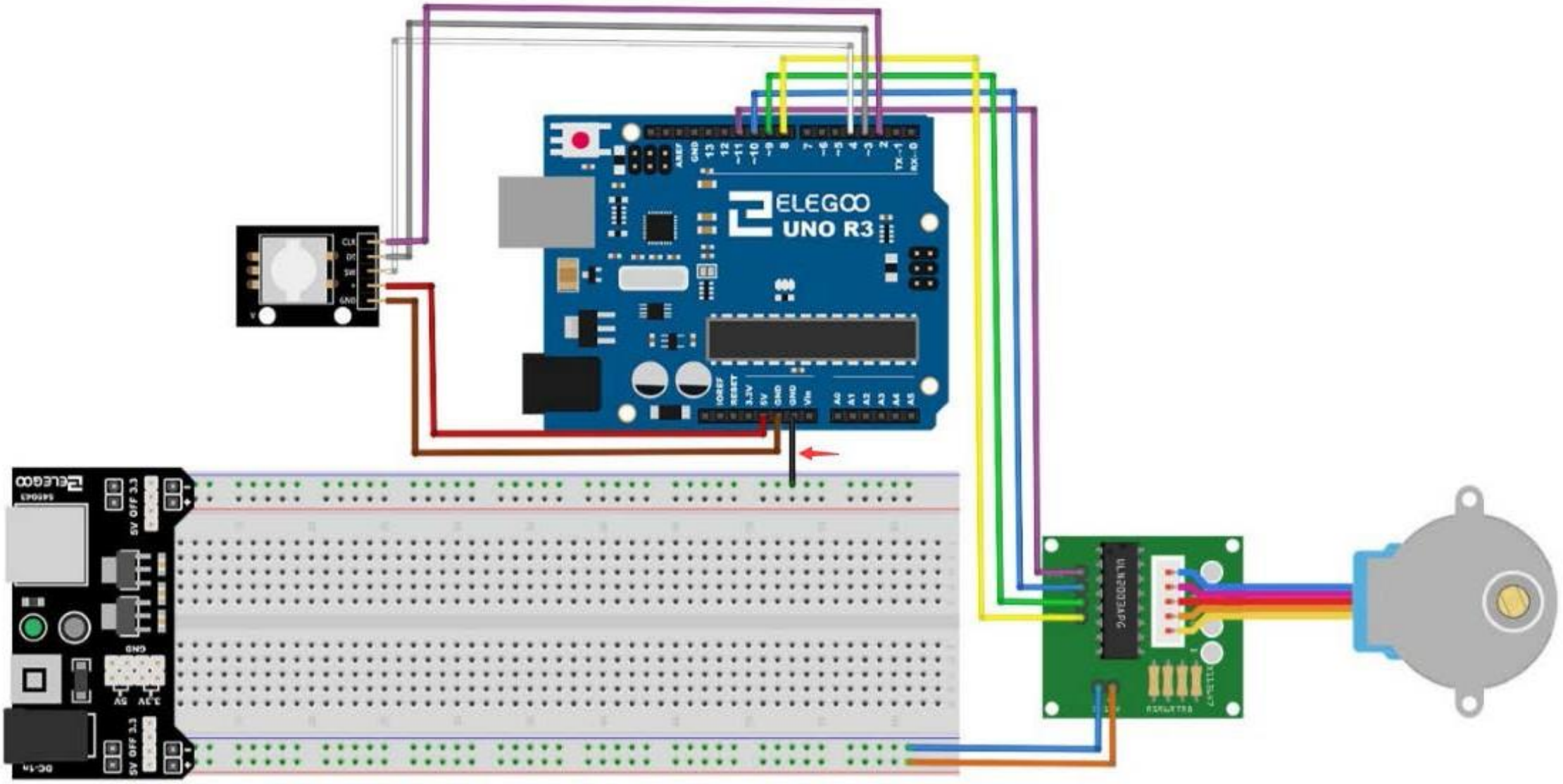


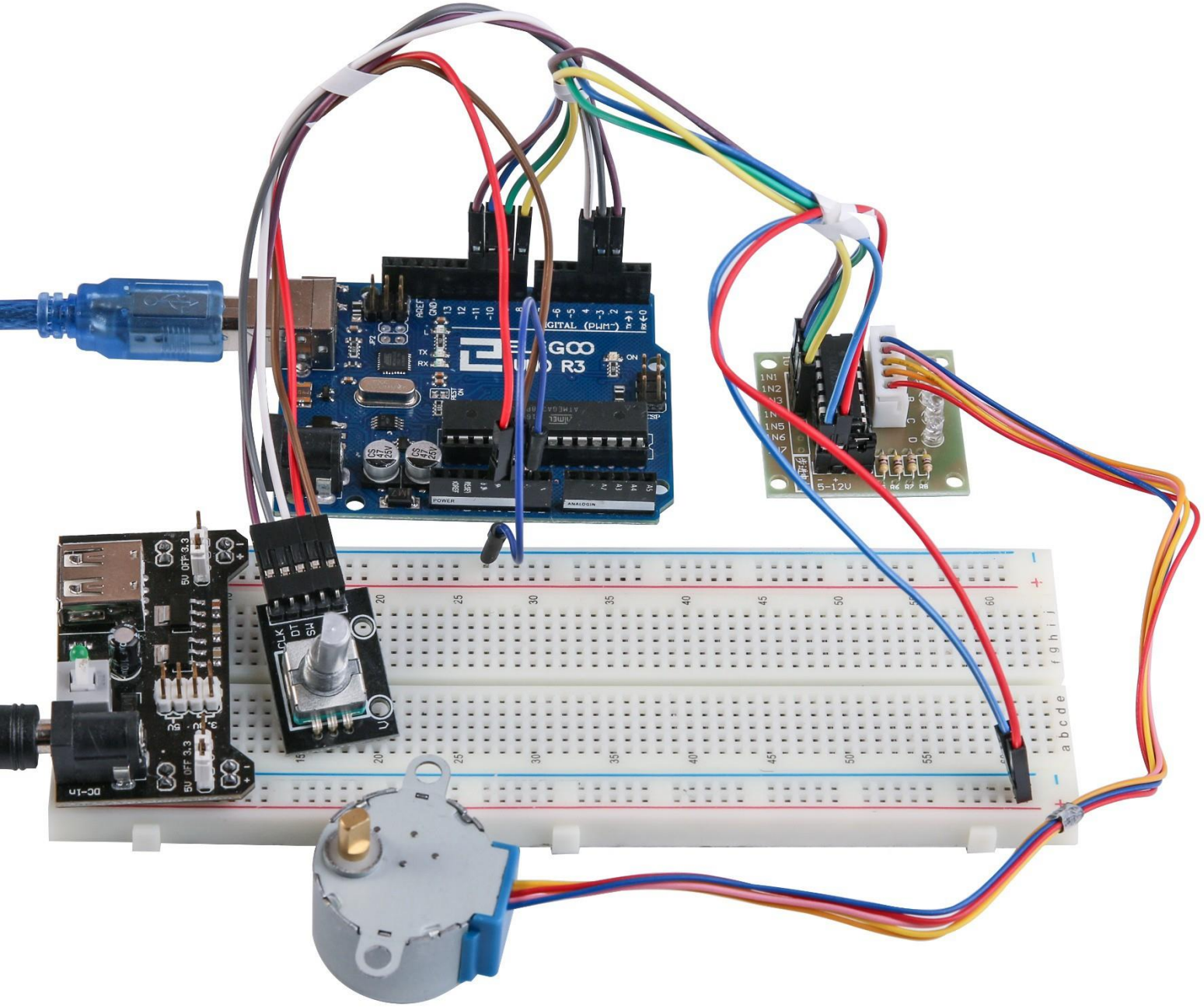
Diagramme de câblage



Nous utilisons 4 broches pour contrôler le Stepper et 3 broches pour le module encodeur rotatif. Les broches 8-11 commandent le moteur pas à pas et les broches 2-4 reçoivent des informations à partir du codeur rotatif.
Nous connectons le 5V et le Ground de l'UNO au codeur rotatif et, par précaution, utilisez une alimentation de la platine pour alimenter le moteur pas à pas, car il peut utiliser plus d'énergie que l'UNO peut fournir.
Nous connectons également UNO Ground à la table de bord pour servir de référence.

Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 33 Controlling Stepper Motor With Rotary Encoder” et téléversez-le sur la carte UNO R3.
Avant toute chose veillez à bien avoir installé la bibliothèque < Stepper > En cas de difficulté, reportez-vous à la leçon 1.

Nous utilisons certaines variables pour stocker la position actuelle, car nous voulons garder une trace de la position du moteur pas à pas afin de pouvoir revenir à la position de départ.
Nous avons également inclus un code de vérification des erreurs pour s'assurer que l'encodeur rotatif ne manque pas les étapes, car cela rendrait notre position du moteur inexacte.

Illustration